THESIS

HABITAT ESTIMATION THROUGH SYNTHESIS OF SPECIES PRESENCE/ABSENCE INFORMATION AND ENVIRONMENTAL COVARIATE DATA

Submitted by
Grant J. Dornan
Department of Statistics

In partial fulfillment of the requirements
for the Degree of Master of Science
Colorado State University
Fort Collins, Colorado
Fall 2011

Master's Committee:

Advisor: Geof H. Givens

Jennifer A. Hoeting
Phillip L. Chapman
Christopher A. Myrick

ABSTRACT

HABITAT ESTIMATION THROUGH SYNTHESIS OF SPECIES PRESENCE/ABSENCE INFORMATION AND ENVIRONMENTAL COVARIATE DATA

This paper investigates the statistical model developed by Foster, et al. (2011) to estimate marine habitat maps based on environmental covariate data and species presence/absence information while treating habitat definition probabilistically. The model assumes that two sites belonging to the same habitat have approximately the same species presence probabilities, and thus both environmental data and species presence observations can help to distinguish habitats at locations across a study region. I develop a computational method to estimate the model parameters by maximum likelihood using a blocked non-linear Gauss-Seidel algorithm. The main part of my work is developing and conducting simulation studies to evaluate estimation performance and to study related questions including the impacts of sample size, model bias and model misspecification. Seven testing scenarios are developed including between 3 and 9 habitats, 15 and 40 species, and 150 and 400 sampling sites. Estimation performance is primarily evaluated through fitted habitat maps and is shown to be excellent for the seven example scenarios examined. Rates of successful habitat classification ranged from 0.92 to 0.98. I show that there is a roughly balanced tradeoff between increasing the number of sites and increasing the number of species for improving estimation performance. Standard model selection techniques are shown to work for selection of covariates, but selection of the number of habitats benefits from supplementing quantitative techniques with qualitative expert judgement.

Although estimation of habitat boundaries is extremely good, the rate of probabilistic transition between habitats is shown to be difficult to estimate accurately. Future research should address this issue. An appendix to this thesis includes a comprehensive and annotated collection of R code developed during this project.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1 Background

The research described in this paper relates to the model for habitat estimation developed by Foster, et al. (2011). They developed a statistical model that uses species presence/absence information, as well as environmental covariate measurements from a set of sampled sites to define both habitat characteristics and habitat boundaries while explicitly accounting for uncertainty. An attractive feature of this approach is that we do not need to restrict our view to looking at one species at a time, nor are we basing our habitat definition solely on the environmental factors of the study region. Rather, this model can incorporate both biological and environmental information to define and locate habitats. Thus, habitats are estimated from more of an ecosystem viewpoint than a physical or spatial viewpoint. The model is described in detail in Section 2.1.

The work I describe here pursues several different avenues of research related to this model. Primarily, I develop and study model performance for several examples, both simple and challenging. Study of these examples is done via Monte Carlo simulation in many cases. These scenarios demonstrate the model's range of flexibility and illustrate methods for the challenging task of parameter estimation. Also, I illustrate specific issues such as the use of the model including sample size effects, model selection, choice of starting parameter values, and convergence of the estimation algorithm. Finally, I present code and documentation used for my research. These items are given in the Appendix.

1.2 Motivation

Suppose that we are asked to advise resource managers and policy makers about the spatial location of species habitats or related ecosystems within a larger spatial domain. For example one might seek to understand the composition of fish species that might be found at various locations throughout the Great Barrier Reef. The distribution of species across locations may be seen as being dependent upon, or as a de facto definition of, habitats.

Imagine that we have access to a data set comprised of presence/absence information for a collection of species observed (or not observed) at a sample of locations throughout the Great Barrier Reef. Further, assume that for each site we can obtain several physical covariates that are relevant to marine life, for instance, salinity, water temperature, pollutant levels, etc. Such covariates are measured only at species sampling sites. The model discussed in this paper allows us to incorporate both types of information into habitat estimation and prediction of species presence probabilities within habitats.

In fact, there is nothing fundamental to the model itself that requires that we study marine habitats, or even habitats at all. We could easily apply this method to animal and/or plant habitats on land without any change to our conceptualization of the model. We could replace species data with mineral presence/absence data and use the model to estimate maps of mineral families across the Rocky Mountains. For simplicity, I will refer to the baseline context of marine habitats characterized by species presence/absence for the remainder of the paper.

There are two types of data used by the model. First we need binary presence/absence data for the species whose joint presence probabilities are used to distinguish our habitats. Second, the model requires covariate data for each of the locations in the sample. It is important to note that these data need not be sampled over some grid. Locations may be chosen by convenience.

It is worth noting from the outset that the two broad questions that we want to address with this model, namely the character and the location of habitats, are actually deeply interrelated. Specifically, if we describe a set of habitats differently in terms of characterizing species presence probabilities, it is natural that the habitat boundaries will be different as well. The converse is equally true.

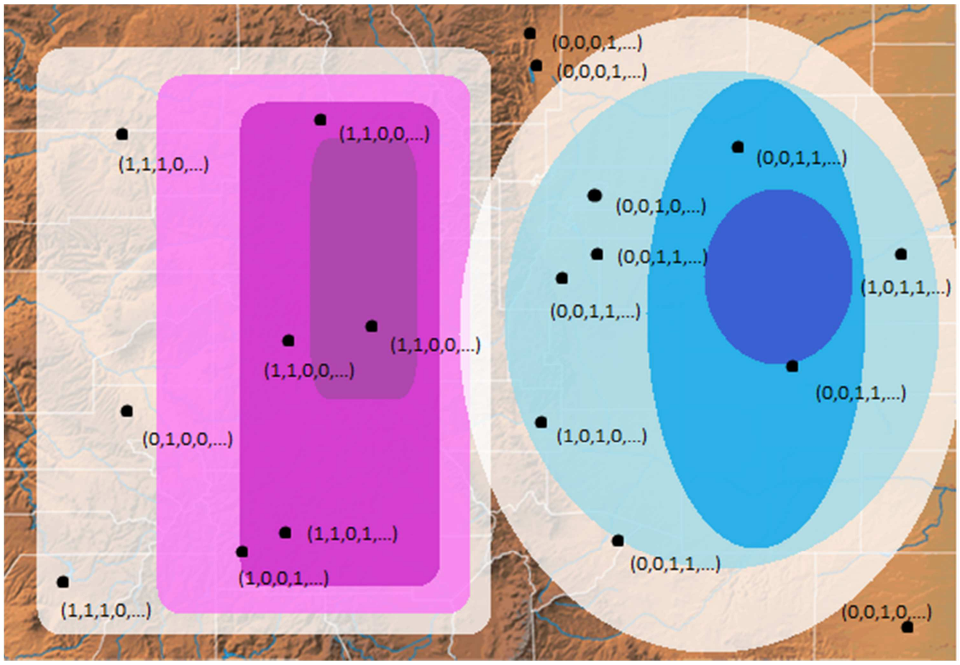

Figure 1.2.1: Colorado Map[1] with two competing habitats and a sample of sites with mock species presence/absence data.

Throughout the exercises described in this paper, plots and maps are indispensible tools in understanding the model. Figure 1.2.1 visually introduces the concept of probabilistic habitats competing over a map of Colorado. We see two habitats, represented by blue in the east and purple in the west. For each, darker colors represent greater probability that the location truly corresponds to that particular habitat. We also see a sample of sites taken from across the state (dots). At each sampled location a binary vector of species presence/absence observations is displayed (1 representing presence, 0 for absence). Sites that share a tendency toward a common habitat are generally expected to share a similar set of species observed. As described mathematically in Section 2.1 and developed through examples in Chapters 3 and 4, the model in Foster et al. naturally allows for habitats to overlap in the sense that we see here in Figure 1.

[1] Underlying map Image found at http://geology.com/state-map/colorado.shtml on August 17, 2011

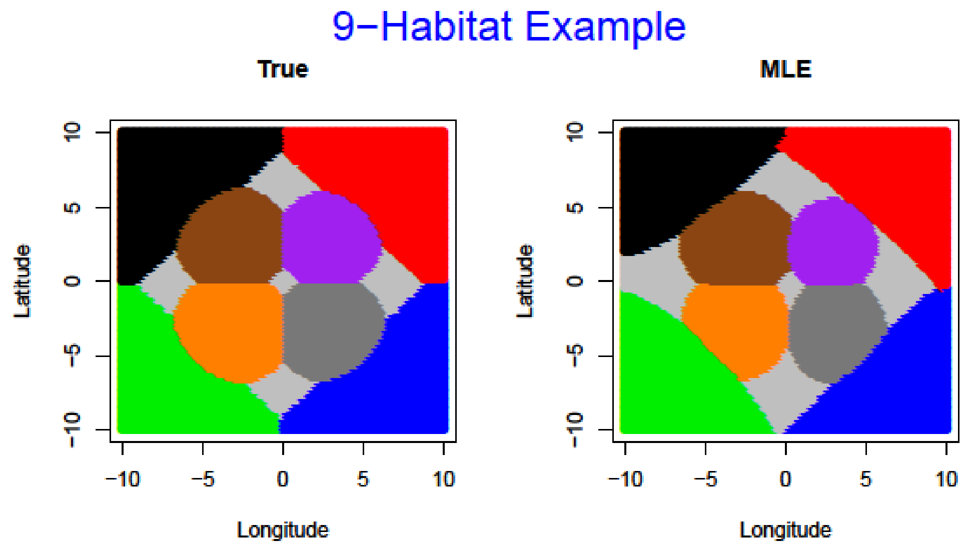

Figure 1.2.2: Example of habitat estimation for a 9 habitat system.

Figure 1.2.2 is an example of the type of habitat scenario that can be estimated with the model and likelihood optimization algorithm described in this paper. While much more explanation is required to fully understand these plots, a cursory description can be understood now. The left-hand plot above displays a study region with 9 color-coded habitats. Through maximum likelihood estimation using the framework of this model, we obtain the estimated habitat map displayed in the right-hand plot. Clearly, extremely good results can be obtained for some analyses.

CHAPTER 2: A STATISTICAL MODEL FOR IDENTIFYING HABITATS

The formal statistical model examined in this paper was developed by Foster et al. (2011). It is essential that a full formulation be provided here as well to understand what follows. Thus in the following section I describe the model which the authors began developing in 2009. The authors have granted me permission to describe their model in language that is, in large part, the same as theirs. Description of my own work will continue again in Section 2.2.

2.1 The Model

Let us begin by considering presence/absence information for *S* species at *n* locations throughout the study region. If at a given site, *i* = 1…*n*, we find the presence of a given species, *j* = 1…*S*, we denote y_{ij} = 1. Conversely, if species *j* is absent at site *i*, we let $y_{ij} = 0$. Thus $\boldsymbol{y}_i = (y_{i1}, \dots, y_{iS})^T$ is the binary vector of presence/absence data for all *j* species at the *i*th site. Next, consider a set of *p* physical covariate measurements at each of the *n* locations in our sample of sites, namely $\boldsymbol{X}_i = (X_{i1}, \dots, X_{ip})^T$. Suppose further that there are *H* habitats in the study region and for the sake of exposition assume that site *i* is a member of habitat *h*. This extra assumption will be relaxed during the development of the model. The goal of this analysis is to incorporate both the presence/absence and covariate information to model presence probabilities and habitat definition.

Temporarily ignoring the information about habitats contained in the covariate data—which will later be used—Foster, et al. argue that the effect of such information is manifested in species distributions. Specifically, a habitat can be defined as a region of environmental space that has approximately constant presence/absence probabilities for each species. Further, these probabilities are distinguishable from those of other habitats. The authors of this model believe that directly linking species distribution to habitat definition is a more defensible approach than simply correlating habitat definition to covariate data.

The model assumptions specify that, within habitat *h*, the probability of detection for the *j*th species is constant. That is to say

$$E\left(y_{ij}\middle|\text{site } i \text{ is in habitat } h\right) = \mu_{jh} = logit^{-1}\left(\alpha_j + \tau_{jh}\right) = \frac{\exp\{\alpha_j + \tau_{jh}\}}{1 + \exp\{\alpha_j + \tau_{jh}\}} \tag{1}$$

where $\mu_{jh} \in [0,1]$ and $\sum_h \tau_{jh} = 0$ for the *j*th species. α_j is the mean presence probability for species *j* across all habitats while τ_{jh} is the habitat specific contribution to species *j*'s presence probability within the *h*th habitat. In accordance with our definition of habitats, there are no terms involving *i* on the right hand side of (1) since species probabilities are assumed to be constant within a (known) habitat. However, that statement is conditional on habitat-knowledge that is, in fact, not known prior to analysis. We allow the probability of a site belonging to each habitat to vary with physical covariates. The changes in these probabilities alter the marginal distribution produced by the model. Until that dependence is specified (below), covariate effects may be viewed as unexplained variation.

Later, when I estimate these model parameters, it will be convenient to group the α_j and τ_{jh} into one parameter matrix. I denote this matrix of species presence probability contributions as ***A***, and define:

$$\boldsymbol{A}_{S\times(H+1)} = \begin{pmatrix} \alpha_1 & \tau_{11} & \cdots & \tau_{1H} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_S & \tau_{S1} & \cdots & \tau_{SH} \end{pmatrix} \tag{2}$$

Now we admit that we do not know which habitat site *i* belongs to. We introduce a random vector for each site, $\boldsymbol{z}_i = (z_{i1}, \ldots, z_{iH})^T$, that identifies site *i* to its unobserved habitat. The elements of any single $\boldsymbol{z}_i$ are all zeros except one element, which equals one in the *h*th position when the *i*th site belongs to the *h*th habitat. It is important to understand from the outset that while we assume that each site truly belongs to a single habitat, we concede that we will never know this relation with certainty. Later we will introduce a model component that allows a smooth probability map for habitats rather than the hard-edged partition implied by the $\boldsymbol{z}_i$.

The model for $\boldsymbol{y}_i$ conditional on habitat type can be expressed as

$$E(\boldsymbol{y}_i|\boldsymbol{z}_i) = \sum_{h=1}^{H} z_{ih}\boldsymbol{\mu}_{\bullet h} \tag{3}$$

where $\text{logit}(\boldsymbol{\mu}_{\bullet h}) = (a_1 + \tau_{1h}, a_2 + \tau_{2h}, \dots, a_S + \tau_{Sh})^T$ using the obvious notation. Note that equations (1) and (3) are equivalent but (3) directly incorporates habitat type.

Recalling that $\boldsymbol{z}_i$ is not observable, the unconditional expectation is required. It is obtained via

$$E(\boldsymbol{y}_i) = E\big(E(\boldsymbol{y}_i|\boldsymbol{z}_i)\big) = E\left(\sum_{h=1}^{H} z_{ih}\boldsymbol{\mu}_{\bullet h}\right) = \sum_{h=1}^{H} \pi_{ih}\boldsymbol{\mu}_{\bullet h} \qquad (4)$$

where the outer expectation is with respect to $\boldsymbol{z}_i$, the inner expectation is with respect to $\boldsymbol{y}_i|\boldsymbol{z}_i$ and $\boldsymbol{\pi}_i = (\pi_{i1}, \dots, \pi_{iH})^T$ is the expectation of $\boldsymbol{z}_i$, where the elements of $\boldsymbol{\pi}_i$ sum to one for any site *i*. This corresponds to the expectation of a standard mixture model (McLachlan & Peel, 2000).

Equation (4) implies that the modeled value of probability of presence for all the species at site *i* is a weighted average of each habitat's modeled probability. The weights are prescribed by the probability of the site belonging to the different habitat groups. The authors feel that this aspect of the model is important as it implies that each site is not deterministically assigned to any one habitat, thereby allowing a smooth map of, in essence, the relative degree to which a site is characteristic of the various habitats. It also allows uncertainty in habitat membership to naturally be incorporated in the model output.

The full distribution of the observations can be completed with further assumptions about the distributions of $\boldsymbol{z}_i$ and $\boldsymbol{y}_i|\boldsymbol{z}_i$. We assume that $\boldsymbol{z}_i$ is a single draw from a multinomial distribution with mean parameters $\boldsymbol{\pi}_i$. For the distribution of $\boldsymbol{y}_i|\boldsymbol{z}_i$, we assume an independent Bernoulli distribution for each species with mean as in (3). Accordingly each of these distribution assumptions will be a source of variability. The unconditional distribution of all species' data at site *i* is

$$f(\mathbf{y}_i; \boldsymbol{\pi}_i, \boldsymbol{\mu}) = \sum_{h=1}^{H} f(\boldsymbol{y}_i | z_{ih} = 1; \boldsymbol{\mu}_{\bullet h}) P[z_{ih} = 1; \boldsymbol{\pi}_i] \qquad (5)$$

Note that (5) describes the presence probabilities at the i^{th} site only, where $\boldsymbol{\mu} = (\boldsymbol{\mu}_{i1}, \dots, \boldsymbol{\mu}_{iH})$ is the $S \times H$ matrix of binomial means. When multiple sites are considered we allow the possibility that habitat membership probabilities depend on the covariates available to delineate the sample locations. This is done by considering a link-linear model

$$\boldsymbol{\pi}_i = g(\boldsymbol{X}_i; \boldsymbol{B}) \tag{6}$$

where $\boldsymbol{B}$ is a matrix of parameters with dimension $p \times (H-1)$ and g maintains the constraint that the elements of $\boldsymbol{\pi}_i$ sum to one for each *i*.

A suitable choice for *g* is the additive logistic function (Aitchison, 1982), which is sometimes referred to as the multinomial logit (Kedem & Fokianos, 2002), and is specified through a model for the site membership probabilities

$$\Pr(z_{ih} = 1 \mid \boldsymbol{X}_i) = \pi_{ih} = \begin{cases} \dfrac{\exp(\boldsymbol{X}_i^T \boldsymbol{B}_{\bullet h})}{1 + \sum_{h=1}^{H-1} \exp(\boldsymbol{X}_i^T \boldsymbol{B}_{\bullet h})}, & if\ 1 \leq h \leq H-1 \\ 1 - \sum_{h=1}^{H-1} \pi_{ih}, & if\ h = H \end{cases} \tag{7}$$

where $\boldsymbol{B}_{\bullet h}$ is the *h*th column of $\boldsymbol{B}$. Note that this implies that the final habitat is defined as the remainder region where all other habitats are unlikely.

Inference for this model follows from the estimation of the B_{kh}, α_j and τ_{jh} parameters.

2.2 Likelihood Function

Having reviewed the model of Foster et al. (2011), I now turn to the tasks of fitting the model through the maximum likelihood method and examination of its performance. This section will develop the likelihood function associated with this model. At this point, and everywhere hereafter, I resume with my own work. The likelihood function for the *i*th site is

$$\begin{aligned} f(\boldsymbol{y}_i; \boldsymbol{\pi}_i, \boldsymbol{\mu}) &= f\left(\begin{bmatrix} y_{i1} \\ \vdots \\ y_{iS} \end{bmatrix}; \begin{bmatrix} \pi_{i1} \\ \vdots \\ \pi_{iH} \end{bmatrix}, \boldsymbol{\mu}\right) = \sum_{h=1}^{H} \left\{ f\left(\begin{bmatrix} y_{i1} \\ \vdots \\ y_{iS} \end{bmatrix} \mid z_{ih} = 1; \begin{bmatrix} \mu_{i1} \\ \vdots \\ \mu_{iH} \end{bmatrix}\right) P\left(z_{ih} = 1; \begin{bmatrix} \pi_{i1} \\ \vdots \\ \pi_{iH} \end{bmatrix}\right) \right\} \\ &= \sum_{h=1}^{H} \left\{ \left[\prod_{j=1}^{S} f\left(y_{ij} | z_{ih} = 1; \mu_{jh}\right) \right] P \left[\boldsymbol{z}_i = (0 \ldots 1 \ldots 0); \begin{bmatrix} \pi_{i1} \\ \vdots \\ \pi_{iH} \end{bmatrix} \right] \right\} . \end{aligned} \tag{8}$$

where the 1 occurs in the *h*th position of the $\boldsymbol{z}_i$ vector in the final expression, and the last equivalence is true by the conditional independence of species within habitat. Using the assumptions that $\boldsymbol{y}_i | \boldsymbol{z}_i$ has an independent Bernoulli distribution and $\boldsymbol{z}_i$ is a single draw from a multinomial distribution with mean parameters $\boldsymbol{\pi}_i$, we can further say that (7) is proportional to

$$\sum_{h=1}^{H}\left\{\left[\prod_{j=1}^{S}\mu_{jh}^{y_{ij}}\left(1-\mu_{jh}\right)^{1-y_{ij}}\right]\pi_{i1}^{z_{i1}}\dots\pi_{ih}^{z_{ih}}\dots\pi_{i(H-1)}^{z_{i(H-1)}}\left(1-\sum_{h^*=1}^{H-1}\pi_{ih^*}\right)^{1-\sum_{h^*=1}^{H-1}z_{ih^*}}\right\}. \tag{9}$$

Because all but one of the z_{ih} (h=1…H) for a given site i is equal to zero, only the $\pi_{ih}^{z_{ih}}$ term corresponding to the hth habitat will differ from $\pi_{i1}^{0}=1$. Moreover, since $z_{ih}=1$ when the ith site belongs to the hth habitat, the term that remains can be reduced to the membership probability, π_{ih}, itself. Finally, the joint likelihood based on all n sites can be written as

$$\prod_{i=1}^{n}\{N_1M_1+N_2M_2+\cdots+N_HM_H\} \tag{10}$$

where

$$N_h=\prod_{j=1}^{S}\mu_{jh}^{y_{ij}}\left(1-\mu_{jh}\right)^{1-y_{ij}}, \qquad h=1\dots H$$

and

$$M_h=\begin{cases}\dfrac{\exp(\boldsymbol{X}_i^T\boldsymbol{B}_{\bullet h})}{1+\sum_{h=1}^{H-1}\exp(\boldsymbol{X}_i^T\boldsymbol{B}_{\bullet h})}, & if\ 1\le h\le H-1\\ 1-\sum_{h=1}^{H-1}\pi_{ih}, & if\ h=H\ .\end{cases}$$

Putting these parts together, the overall likelihood is:

$$\begin{aligned}\prod_{i=1}^{n}\Bigg\{&\left[\prod_{j=1}^{S}\mu_{j1}^{y_{ij}}\left(1-\mu_{j1}\right)^{1-y_{ij}}\right]\frac{\exp(\boldsymbol{X}_i^T\boldsymbol{B}_{\bullet 1})}{1+\sum_{h=1}^{H-1}\exp(\boldsymbol{X}_i^T\boldsymbol{B}_{\bullet h})}+\cdots\\&+\left[\prod_{j=1}^{S}\mu_{j(H-1)}^{y_{ij}}\left(1-\mu_{j(H-1)}\right)^{1-y_{ij}}\right]\frac{\exp\left(\boldsymbol{X}_i^T\boldsymbol{B}_{\bullet(H-1)}\right)}{1+\sum_{h=1}^{H-1}\exp(\boldsymbol{X}_i^T\boldsymbol{B}_{\bullet h})}\\&+\left[\prod_{j=1}^{S}\mu_{jH}^{y_{ij}}\left(1-\mu_{jH}\right)^{1-y_{ij}}\right]\left[1-\frac{\exp(\boldsymbol{X}_i^T\boldsymbol{B}_{\bullet 1})}{1+\sum_{h=1}^{H-1}\exp(\boldsymbol{X}_i^T\boldsymbol{B}_{\bullet h})}-\cdots-\frac{\exp\left(\boldsymbol{X}_i^T\boldsymbol{B}_{\bullet(H-1)}\right)}{1+\sum_{h=1}^{H-1}\exp(\boldsymbol{X}_i^T\boldsymbol{B}_{\bullet h})}\right]\Bigg\}.\end{aligned} \tag{11}$$

In the next section I develop a method to estimate the model parameters via optimization of this likelihood function.

2.3 Estimation Algorithm

I use maximum likelihood estimation to fit the B_{kh}, α_j and τ_{jh} parameters that characterize the modeled habitats and presence/absence probabilities. Right away we notice that this can be a very large estimation problem. The ***B*** parameter matrix is of size $p \times (H-1)$, where *p* is the number of covariates that describe each site. I group the related α_j and τ_{jh} parameters into one matrix, denoted ***A*** (see equation (2) in the previous section)***,*** where the first column contains α_j's and the subsequent *H* columns contain the τ_{jh}'s. There is one row for each species, thus the **A** parameter matrix is of size $S \times (H+1)$. However, since I constrain the sum of the τ_{jh}'s to equal zero, we only need to estimate the first *H* columns of this matrix. All told, we must estimate $p \times (H-1) + S \times H$ parameters. In the case of a realistic Great Barrier Reef data set, this could be more than 4,000 parameters[2].

The method used here employs a blocked non-linear Gauss-Seidel algorithm (Givens & Hoeting, 2005). In other words, the strategy is to optimize our parameter matrices cyclically one set at a time. In particular, I define a block to include one row in the ***A*** parameter matrix or one column in the ***B*** parameter matrix. Then I optimize over that set of parameters while holding the rest fixed at their current value. Since rows correspond to species in the ***A*** matrix and columns to habitats in the ***B*** matrix, a total of $S + H - 1$ individual optimizations must be performed to include all parameters we wish to estimate at this stage. By utilizing this strategy, a potentially very large optimization problem has been broken into many manageable optimization tasks, each of size *p* or *H* parameters. Having completed this process for each row, I repeat the process starting back at the first row in the ***A*** parameter matrix. Algorithm iterations continue like this until convergence is reached. Algebraically, this process can be specified as:

1) Assume $\boldsymbol{B}_0 = \boldsymbol{B}_{0,p\times(H-1)}$ and $\boldsymbol{A}_0 = \boldsymbol{A}_{0,S\times H}$
2) Let *j*=1 and $\boldsymbol{A}_{j\bullet,update} = \arg\max_{\boldsymbol{\alpha\tau}_{j\bullet}} L(\alpha,\tau,B|x,y)$
3) Set $\boldsymbol{A}_{j,1:H} = \boldsymbol{A}_{j\bullet,update}$
4) Let $\boldsymbol{A}_{j,(H+1)} = -\sum_{k=2}^{H}(\boldsymbol{A}_{jk})$
5) Repeat steps 2-4 for *j*=2…*S*

[2] From the beginning of my work with the project, the goal of the development of this model has been to eventually apply it to at least one of three real marine species datasets from Australia. The largest set is Great Barrier Reef data with approximately 1200 sites, 200 species, 13 covariates and 15-20 habitats. Two smaller sets are for a Southeast Australian fishery and a Northwest Australian fishery, each of which having on the order of 120 sites and 100 species. The scenarios developed for this paper keep this motivation in mind, but tackle more manageable examples while investigating the possible effects of increasing *S*, *H* and/or *n*.

6) Let h=1 and $\boldsymbol{B}_{\bullet h,update} = \underset{\boldsymbol{B}_{\bullet h}}{\arg max}\, L(\alpha, \tau, B|x, y)$

7) Set $\boldsymbol{B}_{\bullet h} = \boldsymbol{B}_{\bullet h,update}$

8) Repeat steps 6 and 7 for h=2,…,H-1

9) Repeat steps 2-8 until convergence

10) Final matrices are $\widehat{\boldsymbol{B}}$ and $\widehat{\boldsymbol{A}}$

A coded version of this algorithm can be found in Appendix A.

I considered a relative convergence criterion to halt the algorithm when it appeared to converge (hopefully to the MLEs). This criterion was to stop when the following expression was satisfied:

$$RCC_B = \underset{k,h}{max}[abs(Q_{kh})] < \delta \qquad and \qquad RCC_A = \underset{p,h}{max}\left[abs\left(R_{ph}\right)\right] < \delta$$

where Q_{kh} is the k, h element of $\frac{B_{new} - B_{previous}}{B_{previous} + \varepsilon}$ and R_{ph} is the p, h element of $\frac{A_{new} - A_{previous}}{A_{previous} + \varepsilon}$.

I set $\varepsilon = 10^{-5}$ which ensures against dividing by zero, and δ is set to 10^{-6} for the examples in this paper. However, in practice I typically found that this algorithm converges fairly rapidly and clearly (i.e. in 3 or 4 iterations), so monitoring the RCC was not essential. The topic of convergence will be discussed further in Section 4.3.7.

As seen in Appendix 1, each individual optimization step is done using the optim() function in R. While I experimented with BFGS, conjugate gradients and simulated annealing, I used the Nelder-Mead method for every example displayed in this paper because it provided the most consistent results. The likelihood surface for the model is very complex and high-dimensional, so a more thorough study of optimization techniques would be an interesting topic for further investigation.

CHAPTER 3: SIMULATION STUDIES

3.1 Introduction

In this chapter I will describe simulation studies I used to evaluate the model and better understand its strengths and weaknesses. In Section 3.2, I will introduce some of the simulation-estimation factors that will be investigated later on. Section 3.3 will list and describe each of the habitat scenarios whose estimation results will be presented in Chapter 4. This chapter will additionally serve as a primer for understanding some of the plots that will be relied on heavily to describe these habitat scenarios and the results of estimation.

The first step in the simulation process is to create a testing ('truth') scenario. By scenario, I mean a set of maps of the probabilities of each habitat over spatial and covariate regions, along with species presence probabilities within each habitat. Specifically considering a single habitat, this truth is created by inventing both a set of B_{kh} parameters, which define the probabilistic shape and location of the habitat within the study region, and a set of α_j and τ_{jh} parameters which determine the underlying presence probabilities for each of the species considered within this habitat. The leftmost plot in Figure 3.1.1 is an example of a map for a single habitat under a specific set of characterizing B_{kh} parameters. Red represents areas where this particular habitat has a very high probability of existence and blue indicates very low probability for that habitat. Intermediate colors indicate the transition between these extremes.

The second step of the simulation process here is to generate random data consistent with the assumed 'truth'. First, we must draw a sample of sites from the study region. Then for each site we sample other relevant covariate data and binary species presence/absence information. Note that for my work, there is no difference between considering spatial coordinates versus covariate values because one could imagine a direct mapping from the latter to the former. Therefore, I use the spatial coordinates (longitude and latitude) as the covariate variables hereafter without loss of generality. For the purposes of the examples in this paper, the covariate data are sampled uniformly over the spatial extent of the

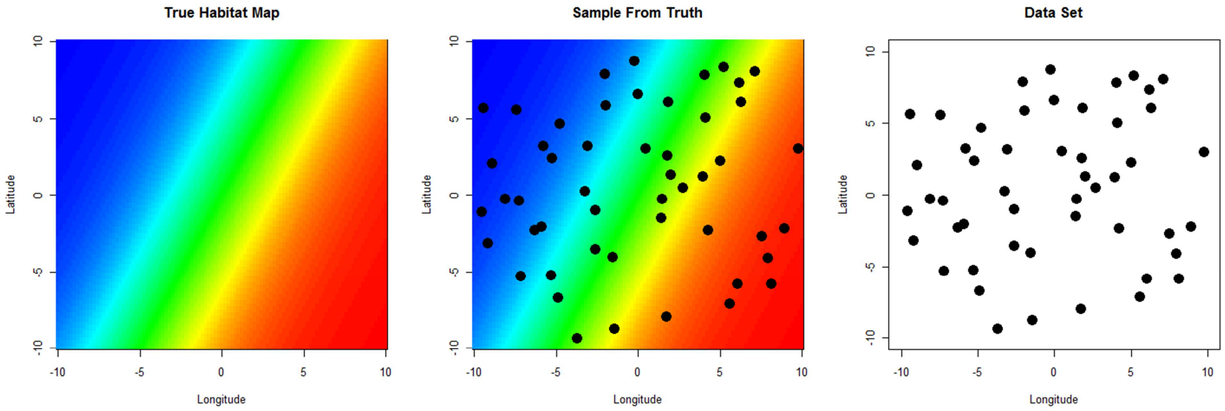

Figure 3.1.1: Illustration of an underlying testing ('truth') scenario.

study region. As previously stated, however, this site sample may be chosen by convenience in practice. The presence/absence data are generated from Bernoulli trials with success probabilities being our species presence probabilities, namely $\mu_{jh} = logit(\alpha_j + \tau_{jh})$ for the j^{th} species in the h^{th} habitat. At this point we 'forget' the truth that we've invented and fit the model to the simulated data to evaluate how well the B_{kh}, α_j and τ_{jh} parameters can be estimated.

It is important to realize that sites are not actually assigned to a particular habitat. The habitat at a specific location is a random variable whose distribution depends on the covariates. Therefore each site is only probabilistically related to each habitat. Section 3.3 will introduce a way to classify sites into specific habitats artificially for the purpose of conceptualization of the scenario, and to provide a useful diagnostic plot to understand model performance.

3.2 Factors Investigated Through Simulation

There are several settings we can adjust (in terms of parameter assumptions, size and complexity of the truth scenario, algorithm execution options, etc.) that affect how the scenario creation, simulation, and estimation processes operate. The following is a list of factors that we must consider as we conduct simulation and Monte Carlo investigation. With a change in some of these factors, we expect a change in our ability to estimate successfully. These effects will be investigated directly in chapter 4 through specific scenarios. Where applicable, the strategy for each factor that is employed throughout the examples in this paper is described.

3.2.1 Quantity and Complexity of True Habitat Maps

Perhaps the most important factor to be decided on in a given problem is the number of habitats, H. With an increase in H, the size of both the $\boldsymbol{A}$ and $\boldsymbol{B}$ parameter matrices grow, thus making the estimation task considerably more demanding. Examples in this paper will contain between three and nine habitats. An interesting topic is misspecification in the number of habitats between truth and model. Section 4.3.5 explores this topic. The potential for using model selection methods to choose the number of habitats will be discussed as well.

Since the $\boldsymbol{B}$ parameter matrix of covariate coefficients defines the habitat maps and we are interested in the types of habitat shapes that can be invented in this framework and estimated by this model, a focal point of my investigation will be to consider several quite distinct true habitat scenarios. For each of the examples in this paper I include just two independent predictors, namely longitude and latitude, denoted X_1 and X_2 respectively. Even while making this simplification, we can still generate interesting habitat maps by introducing polynomial relationships between these two location covariates and habitat probabilities (see Section 3.3). Presumably when using this model with real data, we would have many more covariates and covariates that have their own underlying spatial structure.

3.2.2 Random Sample of n Sites

For each of the examples in this paper we sample sites uniformly across the study region. Unsurprisingly, the number of sites in the sample is one of the biggest factors in our ability to estimate effectively. Chapter 4 will present results pertaining to the effect that sample size has on estimation.

We can easily visualize a situation where a random sample of sites would omit key locations, for instance near habitat boundaries. For this reason, it will also be important to investigate whether site location variability affects estimation. In section 4.3.2 I present the results of Monte Carlo simulation to further understand this variability.

3.2.3 Number of Species

The number of species, S, provides an interesting feature in this model because with an increase in S we gain more data as we simultaneously increase the number of parameters that need to be estimated. An examination of the effect of S on estimation, and specifically the balance between the number of sites and the number of species, will be explored in Section 4.3.3.

3.2.4 Choice of Species Prevalences in each Habitat

Recall that α_j and τ_{jh} respectively represent the grand mean and habitat-specific adjustment for the probability of presence for species j. Unless otherwise noted in this paper, we adopt the assumption that α_j=0 for all j. Given the logit transformation we use to obtain presence probabilities μ_{jh}, this dictates that the average presence probability across habitats is 0.5 for every species. The power of each species to help discriminate among habitats is therefore isolated in the τ_{jh} parameters.

Presumably, habitats are easier to distinguish if less common species in a given habitat are very rare and more common species are very common. Based on this notion, we expect estimation to be easier when the magnitudes of the τ_{jh} are large. Note that if τ_{jh}=log(x), the odds of finding species j in habitat h increase by a factor of x relative to the baseline effect of α alone. Likewise, if τ_{jh}=-log(x), those odds decrease by a factor of x. Unless otherwise stated in this paper we will use ±log(3) for τ values, thus making species presence probabilities 0.25 and 0.75. For simplicity, we draw randomly from these two values of τ when creating the truth from which to simulate data. Note that because of the constraint that the τ's must sum to zero, the last habitat's specific probability effect need not be ±log(3). An investigation comparing various values of τ will be conducted in Section 4.3.6.

3.2.5 A Note on Model Selection

Because the addition of either habitats or covariates (including polynomial coefficients, interactions, etc.) to the model increases the number of total parameters to estimate, we may, in theory, use standard model selection methods to evaluate the inclusion set of either. Doing model selection on covariates is the more familiar of the two topics, and will be investigated in Section 4.3.4. A study on misspecification of the number of habitats can be found in Section 4.3.5.

Model selection may not be feasible in terms of computation time for large scale problems. The topic of computation time will be addressed throughout Chapter 4.

3.2.6 Choice of Starting Values for Optimization

Finding an appropriate and successful strategy for choosing parameter starting values to be fed into the optimization algorithm used to find the MLE has proven to be a very challenging topic. Many different strategies were tested and some are compared in Section 4.3.8. Unless otherwise noted, a single, consistent method for choosing initial values will be assumed. Starting values, $A_{0,jh}$ and $B_{0,kh}$, for the two parameter matrices are selected by drawing randomly from the following two distributions,

$$A_{0,jh} \sim Normal\left(A_{jh}\, , abs(\frac{A_{jh}}{2})\right) \qquad and$$

$$B_{0,kh} \sim Normal\left(B_{kh}\, , abs(\frac{B_{kh}}{2})\right).$$

Thus the signal-to-noise ratio (SNR) for the generation of starting values with respect to the true values is set at 2. Of course, such an approach is not possible in real-world applications. An approach like random starts local search might be useful (Givens & Hoeting, 2005). Later we compare different SNRs against one another and to uninformed starting values to understand better how the choice of starting values affects optimization.

3.3 Testing Scenarios

In keeping with the motivation for this paper, this section will list and display a collection of habitat scenarios that I will later attempt to estimate. As described in Section 3.2, the methods for choosing true values for α_j, τ_{jh} and covariates are consistent throughout the examples presented (unless otherwise stated). I also assume a set of B_{kh} parameters that define the shapes and locations of the habitats. In this paper I consider seven different testing scenarios, each of which I introduce with the defining ***B*** parameter matrix and illustrate with habitat probability plots.

The description of a given scenario requires two last pieces of information, namely the sample of *n* sites across the study region and the simulation of species presence/absence for *S* species at each site. Scenarios listed in this paper employ between 150 and 500 sites, and between 15 and 50 species. While I explore the choice of *n* and *S* in Chapter 4, this section will investigate the specification of the B_{kh} parameters through the resultant habitat probability plots.

Scenario #1: Linear Baseline

The first scenario includes three habitats and linear probabilistic boundaries for each individual habitat. In order to set up a three-habitat example, I only need to explicitly define the first two habitats. A simple, linear, three habitat scenario is constructed by the following ***B*** matrix:

$$\boldsymbol{B}^{\mathrm{T}} = \begin{bmatrix} 0 & 0.4 & -0.2 \\ 0 & 0 & 1 \end{bmatrix}.$$

This scenario will be referred to as the baseline case throughout the paper. We will explore many of the features and results of this model through the lens of this baseline example. Estimation for this scenario is based on a sample of *n*=200 sites and *S*=20 species, results are presented in Section 4.2.1.

It is important to understand how these B_{kh} parameters combine with the covariate data, ***X***, to generate a set of habitat probabilities across the study region. The first row of $\boldsymbol{B}^T$ defines the first habitat in the system (I present $\boldsymbol{B}^T$ rather than ***B*** only to save space). In this linear case, the three elements in this row of $\boldsymbol{B}^T$ correspond, in order, to an intercept contribution, a linear longitude contribution, and a linear latitude contribution to the probability that Habitat 1 exists at a given location in our study region. Denoting longitude and latitude as X_1 and X_2, the i^{th} row of the design matrix is

$$\boldsymbol{X}_i^{\mathrm{T}} = (1 \quad X_1 \quad X_2)\,.$$

Figure 3.3.1 illustrates the habitats defined by ***B*** over a square study region with side lengths of 20 units, centered at the origin. For simplicity, this study region will remain constant throughout the paper. The plots seen here are actually constructed by dividing the study region into a 101×101 grid, taking the covariate (location) data in each subregion, and computing a set of probabilities for each grid point.

For now we will inspect habitat probability plots that are generated while ignoring any other habitats in the system. Specifically, the set of probabilities displayed in the plots in Figure 3.3.1 are calculated in the following way:

$$\pi_{ih} = logit^{-1}\left(\boldsymbol{X}_i^T \boldsymbol{B}_{\bullet h}\right), \qquad h = 1 \dots (H-1)$$

Habitat probability plots whose probabilities are derived from the standard logistic model—expressed separately from the other habitats in the system—will henceforth be referred to as ‘individual logistic habitat probability plots/maps’. As described in Section 3.1, red regions correspond to probability near

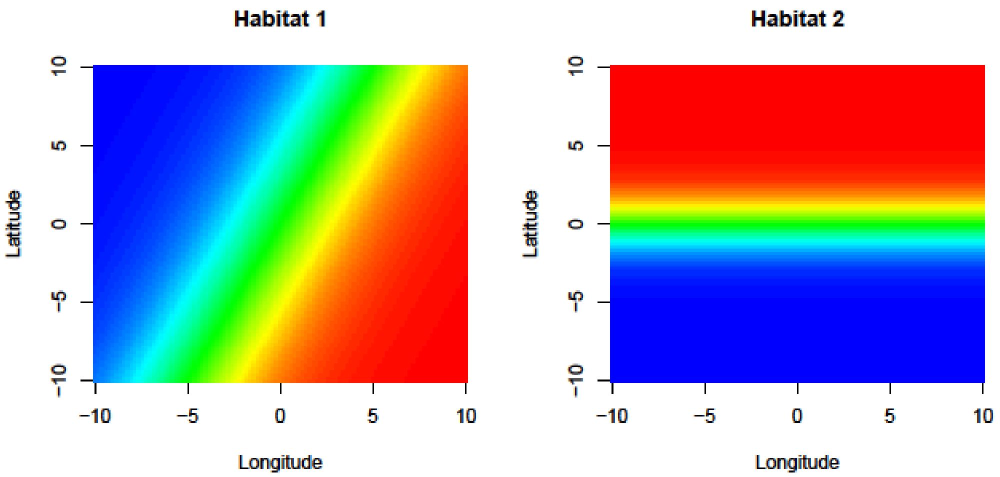

Figure 3.3.1: Individual logistic habitat probability maps for Scenario #1 (Baseline Linear).

1, blue to probability near 0, and intermediate colors to intermediate probabilities of habitat *h* existing at a given location. Note that both of the defined habitats coexist over the same study region. The third habitat, which is defined as the absence of the first two habitats, also exists over this same region.

These plots, illustrate how a vector of parameters, $\boldsymbol{B}_{\bullet h}$ directly define the membership probabilities of the *h*th habitat across the study region, separately from the other habitats. Thus, we have not yet employed the feature of the model that manages habitat intersection and competition. A challenging topic in the early part of my research was how to take these two autonomous sets of *B* parameters and corresponding pictures, and think about the system of habitats holistically. Mathematically, it is the additive logistic aspect of the model that allows us to require the probabilities of habitat membership to sum to one across all habitats for a given site.

The additive logistic model is also the mechanism by which we say that the third habitat is the absence of the first two. Figure 3.3.2 illustrates all three habitats after this transformation. Imagine stacking the three plots of Figure 3.3.2 on top of each other. The additive logistic model ensures that the three habitat membership probabilities at any single location will sum to one. More explicitly, the probabilities mapped in these plots are given by the π_{ih} (h=1…H) defined by equation (7) in Section 2.1. I will refer to these images as ‘additive logistic habitat probability plots/maps’ because these three pictures become dependent upon one another through the additive logistic model.

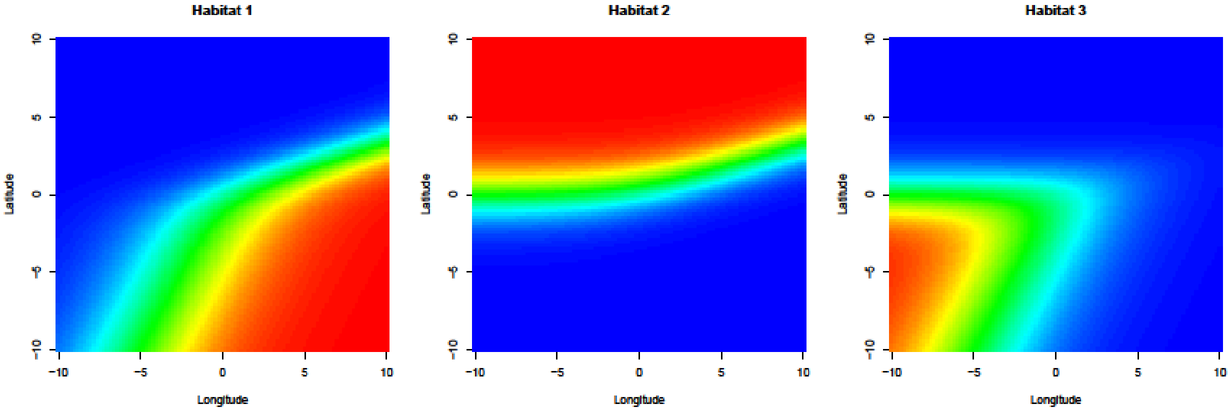

Figure 3.3.2: Additive logistic habitat probability plots for Scenario #1 (Baseline Linear).

For example, we can see in Figure 3.3.1 that sites in the top right corner of the map have high individual logistic membership probabilities for both Habitat 1 and Habitat 2. This competition is echoed in the additive logistic habitat probability plots where these habitats seem to ‘bend away’ from one another in the top-right corner of the plots. Habitat 3 can be found exactly where we should expect it, in the bottom left corner where neither Habitat 1 nor Habitat 2 have high probability.

It will not always be necessary to inspect these additive logistic habitat probability plots, but they are useful in many cases. The individual logistic habitat probability plots are a good visualization of the actual ***B*** parameters, while the additive logistic habitat probability plots more accurately represent what this model tries to emulate.

In the interest of parsimonious results, it would be great to have a way to summarize the *H* additive logistic habitat probability maps into a single image. To this end, I developed a plot that displays the most probable habitat across a grid of locations in the study region (these are not sampled sites). The method for choosing the most probable habitat at a given site is simple: each point on the grid is assigned to the habitat for which the plurality of additive logistic probability at that point is attributed. This graph will be referred to as the habitat classification plot, and for the Baseline Linear example, is displayed in Figure 3.3.3. Black represents the region where habitat one is attributed to locations on the grid, while blue does so for habitat two and red for habitat three.

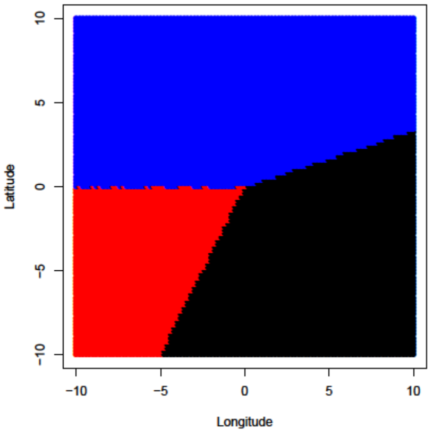

Figure 3.3.3: Habitat classification plot for Scenario #1 (Baseline Linear).

Because of its simplicity, it will often be expedient to look to the habitat classification plot as the chief determination of estimation results. However, we must not forget that there is an uncertainty structure that underlies this hard-edged plot. Two different sets of individual logistic habitat probability plots could in fact lead to identical classification plots (see Section 4.3.9).

Scenario #2: Two Circular Habitats

This second scenario also includes three habitats but employs linear and quadratic effects in both X_1 and X_2. The model is specified by the following matrices:

$$\boldsymbol{B}^{\mathrm{T}} = \begin{bmatrix} -12.75 & 5.10 & -0.51 & 5.10 & -0.51 \\ -12.75 & -5.10 & -0.51 & -5.10 & -0.51 \end{bmatrix} \qquad \boldsymbol{X}_i^{\mathrm{T}} = (1 \quad X_1 \quad X_1^2 \quad X_2 \quad X_2^2)$$

By introducing a quadratic term for each of the location covariates (in the third and fifth columns of $\boldsymbol{B}^T$) we can start to include more complex probability surfaces than planes. The individual logistic habitat probability plots in Figure 3.3.4 show a circular island of high probability for each of the two defined habitats. Estimation results for this scenario are based on a sample of *n*=150 sites and *S*=15 species, and are presented in Section 4.2.2.

The corresponding additive logistic habitat probability plots, are shown in Figure 3.3.4. Since there is no overlap of high probability areas of Habitat 1 and Habitat 2, these additive logistic habitat probability plots are not particularly helpful for understanding how the three habitats coexist. For the examples I study in this paper whose first *H*-1 habitats don't overlap in areas of high probability, I will often omit these additive logistic probability maps.

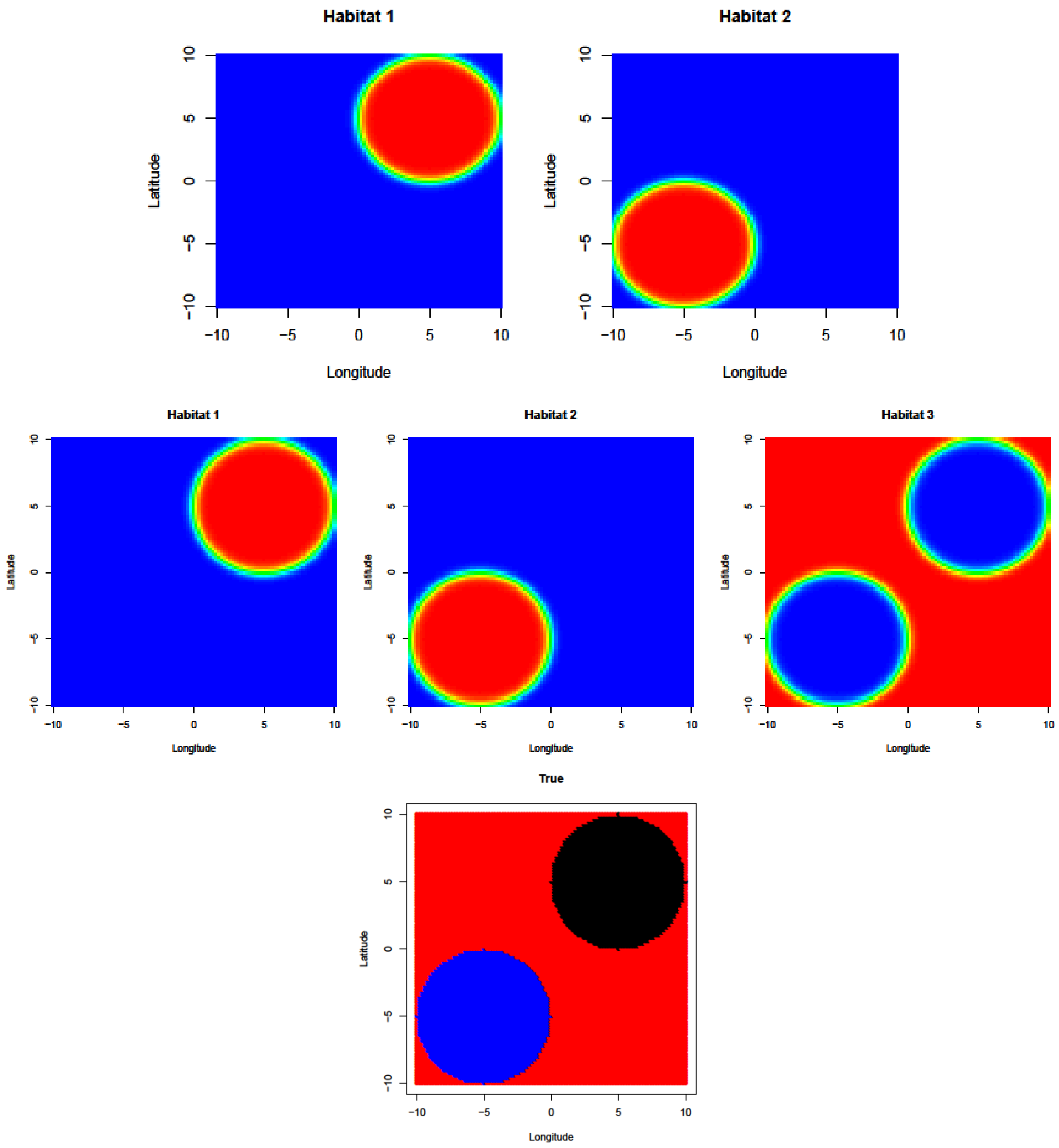

Figure 3.3.4: Individual logistic habitat probability plots (top), additive logistic habitat probability plots (middle) and habitat classification plot (bottom) for Scenario #2 (Two Circles).

Notice the shape of the third habitat in the additive logistic habitat probability plots. I think it is impossible for a set of parameters conforming to the quadratic polynomial structure of ***B*** to result in a habitat with this shape, defined directly. In other words, it is possible that only the first two habitat shapes could be parameterized (recall that the third habitat is defined in the model as the absence of the other two habitats). In fact, this feature proves to be true more often than not in the examples listed in this paper. The ramifications of this idea will be explored in Chapter 4.

Another difference between Scenario #1 and Scenario #2 is the sharpness of the habitat boundaries. In the three-dimensional space where this probability surface exists, we should think of this as the steepness of the probability surface. This steepness can easily be adjusted for a habitat by multiplying the corresponding row of parameters in $\boldsymbol{B}^T$ by a scalar. A scalar multiple that is less than one makes boundaries broader (i.e. less steep), while a scalar greater than one creates steeper boundaries that appear sharper in these two-dimensional habitat probability plots.

Scenario #3: Cubic

This scenario employs linear, quadratic and cubic effects in the covariates (X_1 and X_2 denoting longitude and latitude). The model is specified by the following parameter matrix and (partial) design matrix:

$$\boldsymbol{B}^{\mathrm{T}} = \begin{bmatrix} 10 & 10 & -0.4 & -0.2 & 40 & -0.07 & -0.7 \\ -15 & -10 & 0.4 & 0.2 & -2 & 0.3 & 0.08 \end{bmatrix} \qquad \boldsymbol{X}_i^{\mathrm{T}} = (1 \quad X_1 \quad X_1^2 \quad X_1^3 \quad X_2 \quad X_2^2 \quad X_2^3)$$

Estimation results for this scenario are based on a sample of n=150 sites and S=15 species, and are presented in Section 4.2.3.

Figure 3.3.5 shows the habitat plots for this scenario. By inspecting the additive logistic habitat probability plots, we can see that Habitat 1 is the most dominant habitat. I claim this because after using the additive logistic model, Habitat 1 ‘wins’ the areas that are contested by Habitat 2, thus retaining a similar shape to its individual habitat region. Habitat 3 is oddly shaped and divided across three separate areas.

In the classification plot (Figure 3.3.5), we see that even with the apparently simple framework of only two covariates with polynomial expressions we can achieve complex habitat maps. This cubic scenario naturally lends itself to an ecological interpretation of prime habitat (black), fringe habitat (red), and non-habitat (blue) regions. For example, a collection of fish living on a coral reef may find ideal water temperature, salinity and vegetation in the prime habitat region, but only two of those three characteristics in the fringe habitat and zero or one in the non-habitat region.

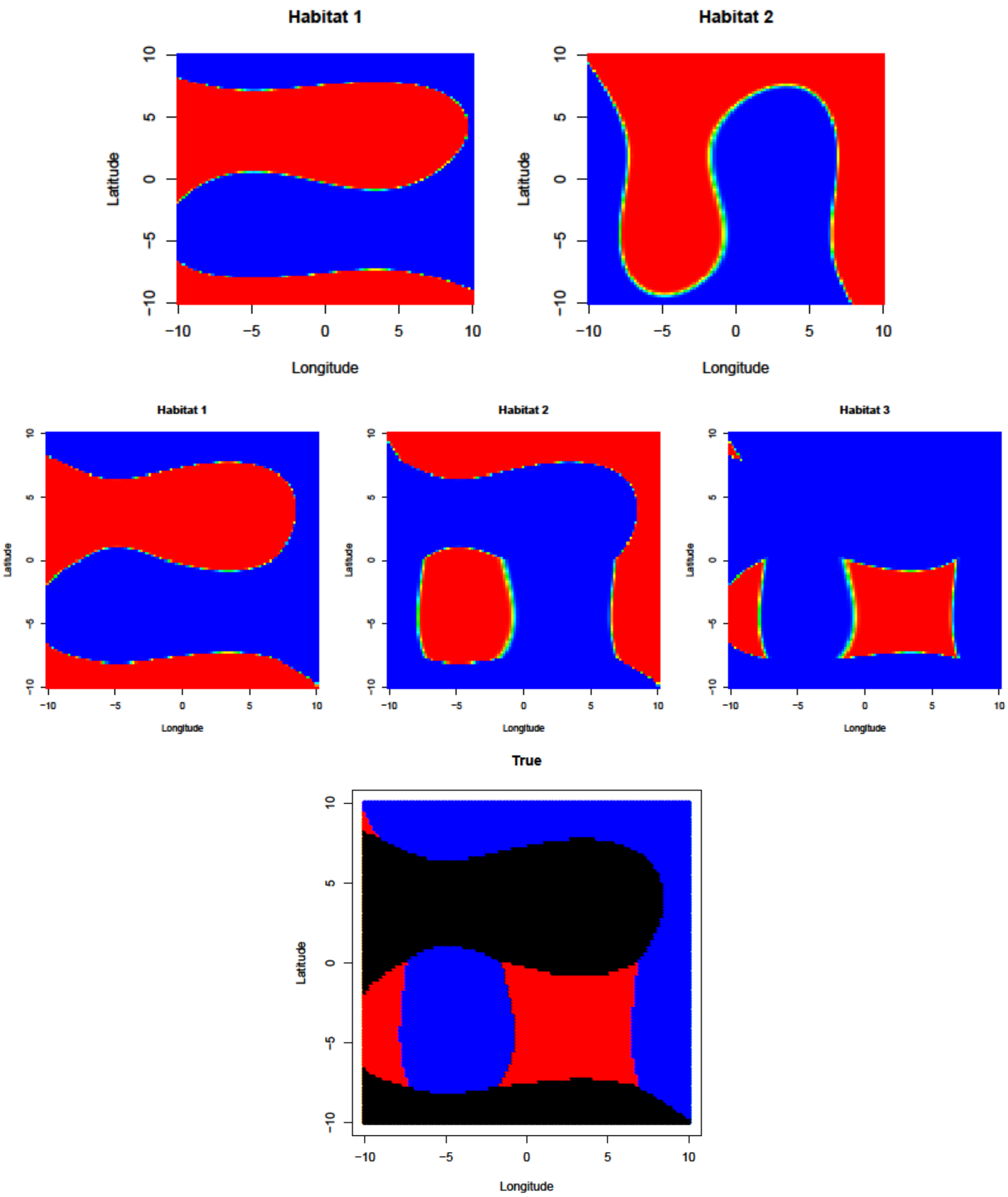

Figure 3.3.5: Individual logistic habitat probability plots (top), additive logistic habitat probability plots (middle) and habitat classification plot (bottom) for Scenario #3 (Cubic).

Scenario #4: Linear/Quadratic/Cubic

This scenario is the first to include four habitats, generated by a third row of parameters in the $\boldsymbol{B}^T$ matrix. Within this system, I incorporate linear, quadratic and cubic individual logistic habitat boundaries (recall these are only probabilistic transition between habitats). This scenario is specified by the following ***B*** matrix and the same design matrix as used in Scenario #3:

$$\boldsymbol{B}^{\mathrm{T}} = \begin{bmatrix} -14.85 & -1.98 & 0 & 0 & -0.40 & 0 & 0 \\ 4.46 & -0.30 & -2.38 & 0.18 & 0.59 & -1.19 & 0.24 \\ -34.20 & 7.63 & -0.55 & 0 & -7.63 & -0.55 & 0 \end{bmatrix}$$

$$\boldsymbol{X}_i^{\mathrm{T}} = (1 \quad X_1 \quad X_1^2 \quad X_1^3 \quad X_2 \quad X_2^2 \quad X_2^3)$$

Notice that the first row of $\boldsymbol{B}^{\mathrm{T}}$, having zeroes in the quadratic and cubic coefficients, generates the linear habitat boundary seen in the top-leftmost panel of Figure 3.3.6. Figure 3.3.6 displays the individual logistic habitat probability plots and the habitat classification plot defined by $\boldsymbol{B}$. I choose not to display the additive logistic habitat probability plots here because the three parameter-defined habitats are clearly disjoint.

In Section 4.2.4 I present estimation results based on a sample of *n*=225 sites and *S*=24 species. Then, in Section 4.3.4, I will use this scenario to investigate model selection based on covariate polynomial order.

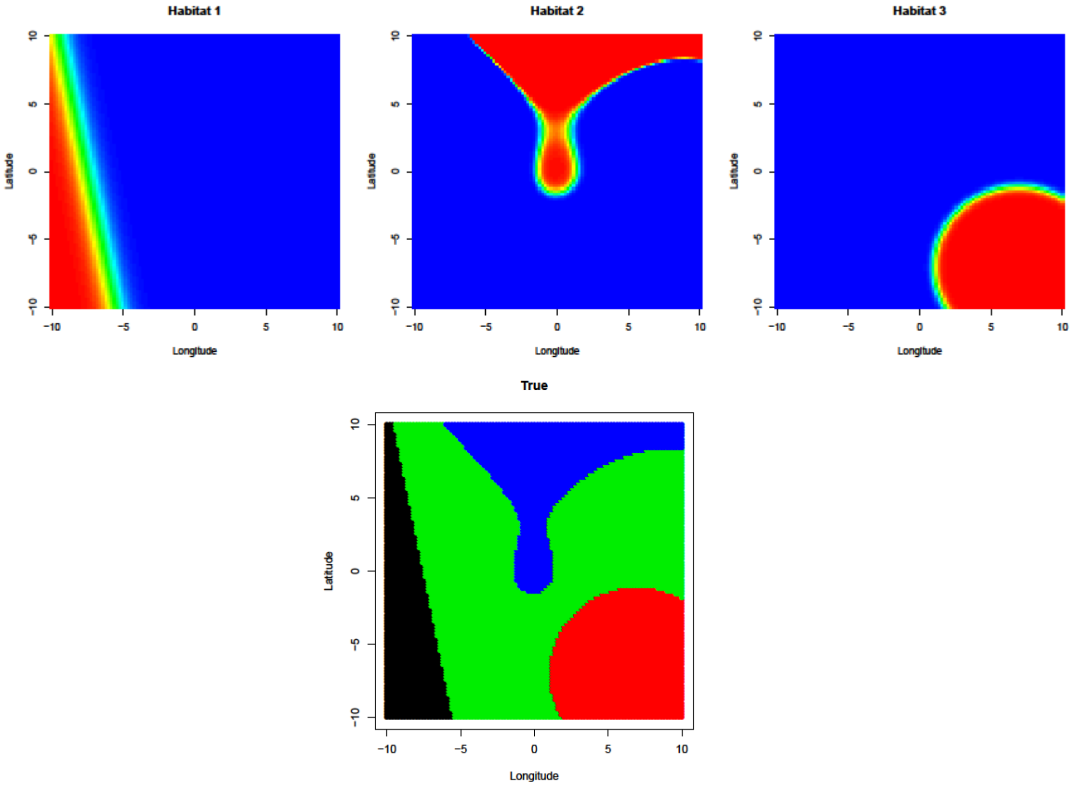

Figure 3.3.6: Individual logistic habitat probability plots (top) and habitat classification plot (bottom) for Scenario #4 (Linear/Quadratic/Cubic).

Scenario #5: Diamond

This scenario includes five habitats and employs only linear effects in the longitude and latitude covariates (X_1 and X_2). The model is specified by the following matrices:

$$\boldsymbol{B}^{\mathrm{T}} = \begin{bmatrix} -10.69 & -1.19 & 1.19 \\ -10.69 & 1.19 & -1.19 \\ -10.69 & 1.19 & 1.19 \\ -10.69 & -1.19 & -1.19 \end{bmatrix} \qquad \boldsymbol{X}_i^{\mathrm{T}} = (1 \quad X_1 \quad X_2)$$

Figure 3.3.7 shows the associated individual logistic habitat probability plots and the habitat classification plot. Estimation results for Scenario #5 (Diamond) are based on a sample of *n*=200 sites and *S*=20 species, and are presented in Section 4.2.5. Additionally in Section 4.2.5 I will fit both a linear form model and a quadratic form model to this scenario to examine the possible effect of model misspecification.

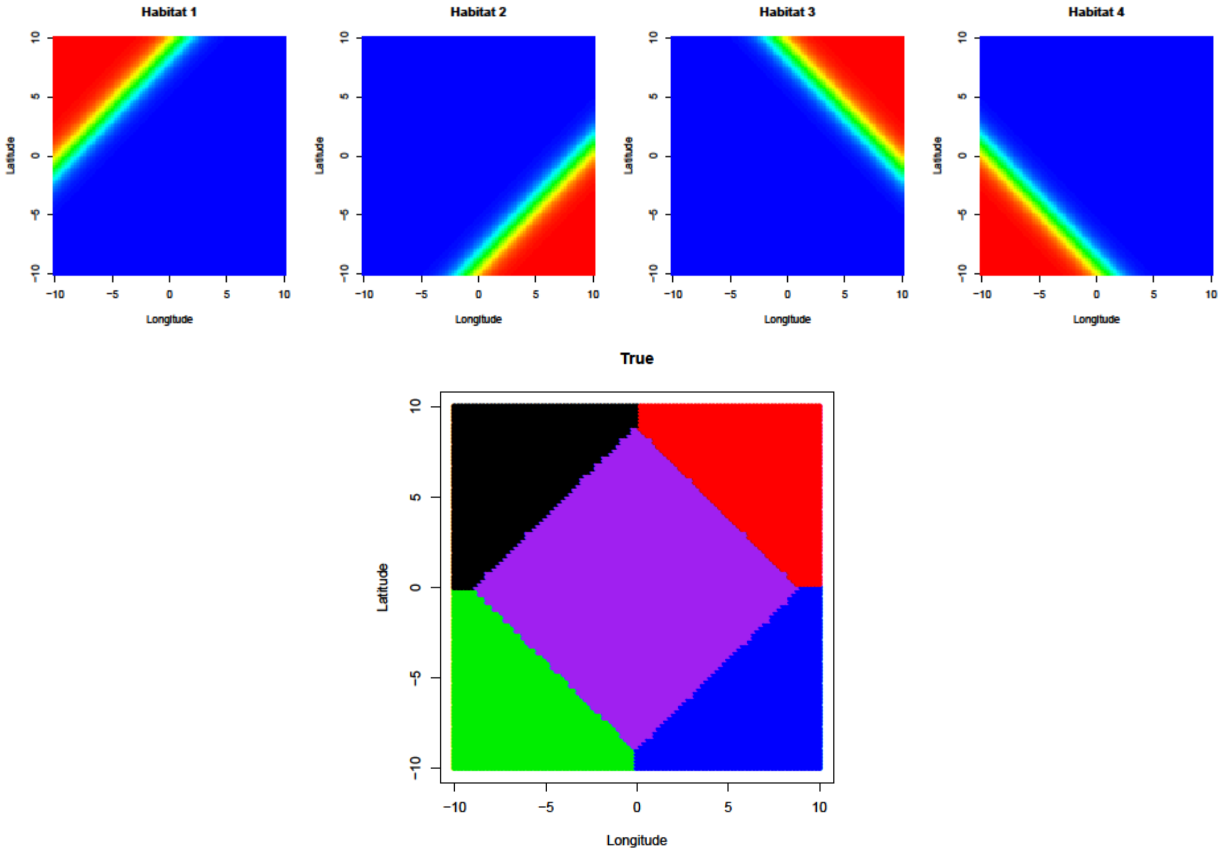

Figure 3.3.7: Individual logistic habitat probability plots (top) and habitat classification plot (bottom) for Scenario #5 (Diamond).

Scenario #6: Four Circles

This scenario includes four circular habitats (plus the fifth remainder region) by employing linear and quadratic effects in X_1 and X_2. The model is specified by the following matrices:

$$\boldsymbol{B}^{\mathrm{T}} = \begin{bmatrix} -12.75 & 5.10 & -0.51 & 5.10 & -0.51 \\ -12.75 & -5.10 & -0.51 & -5.10 & -0.51 \\ -12.75 & -5.10 & -5.10 & 5.10 & -5.10 \\ -12.75 & 5.10 & -5.10 & -5.10 & -5.10 \end{bmatrix} \qquad \boldsymbol{X}_i^{\mathrm{T}} = (1 \quad X_1 \quad X_1^2 \quad X_2 \quad X_2^2)$$

In this scenario, the four circular regions are defined by separate quadratic functions of the covariates. The associated individual logistic habitat probability plots and the habitat classification plot can be seen in Figure 3.3.8. Estimation results for Scenario #6 (Four Circles) are based on a sample of *n*=400 sites and *S*=30 species, and are presented in Section 4.2.6. An analysis of the effect of misspecifying the number of habitats (*H*) will be performed in Section 4.3.5 using this scenario.

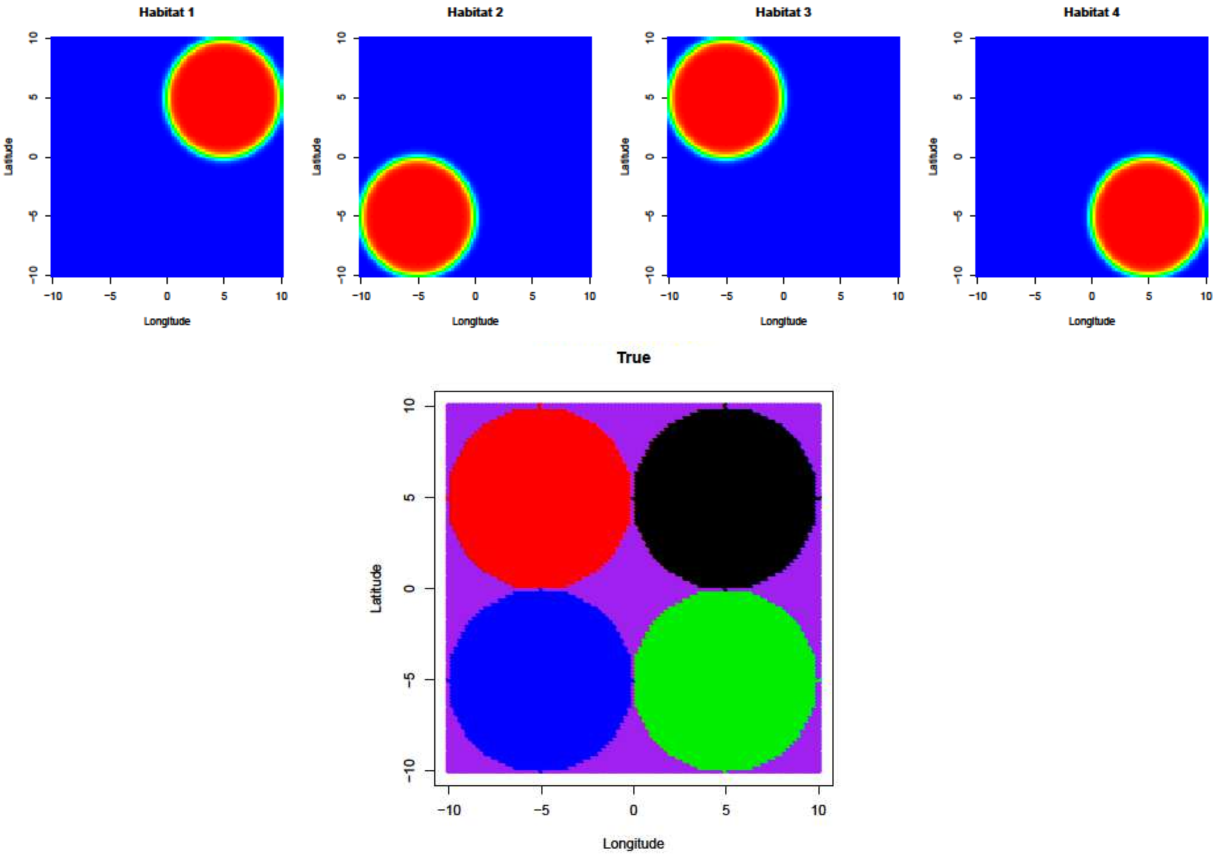

Figure 3.3.8: Individual logistic habitat probability plots (top) and habitat classification plot (bottom) for Scenario #6 (Four Circles).

Scenario #7: Diamond/Circles

This scenario includes nine total habitats with four linear boundaries, four quadratic boundaries and the remainder region. This scenario is modeled with the following matrices:

$$\boldsymbol{B}^{\mathrm{T}} = \begin{bmatrix} -10.69 & -1.19 & 0 & 1.19 & 0 \\ -10.69 & 1.19 & 0 & -1.19 & 0 \\ -10.69 & 1.19 & 0 & 1.19 & 0 \\ -10.69 & -1.19 & 0 & -1.19 & 0 \\ -2.84 & 2.57 & -0.43 & 2.57 & -0.43 \\ -1.44 & -2.57 & -0.43 & -2.57 & -0.43 \\ -2.14 & -2.57 & -0.43 & 2.57 & -0.43 \\ -2.14 & 2.57 & -0.43 & -2.57 & -0.43 \end{bmatrix} \qquad \boldsymbol{X}_i^{\mathrm{T}} = (1 \quad X_1 \quad X_1^2 \quad X_2 \quad X_2^2)$$

Estimation results for Scenario #7 (Diamond/Circles) are based on a sample of *n*=400 sites and *S*=40 species, and are presented in Section 4.2.7. This scenario is potentially challenging since it requires 400 specifying parameters, however we will see in Section 4.2.7 that estimation performance is actually quite good. Figure 3.3.9 shows the eight individual logistic habitat probability plots defined by the above $\boldsymbol{B}^{\mathrm{T}}$. In Figure 3.3.10, I display both the additive logistic habitat probability plots and the habitat classification plot for this scenario. These plots demonstrate that this scenario is large and complex.

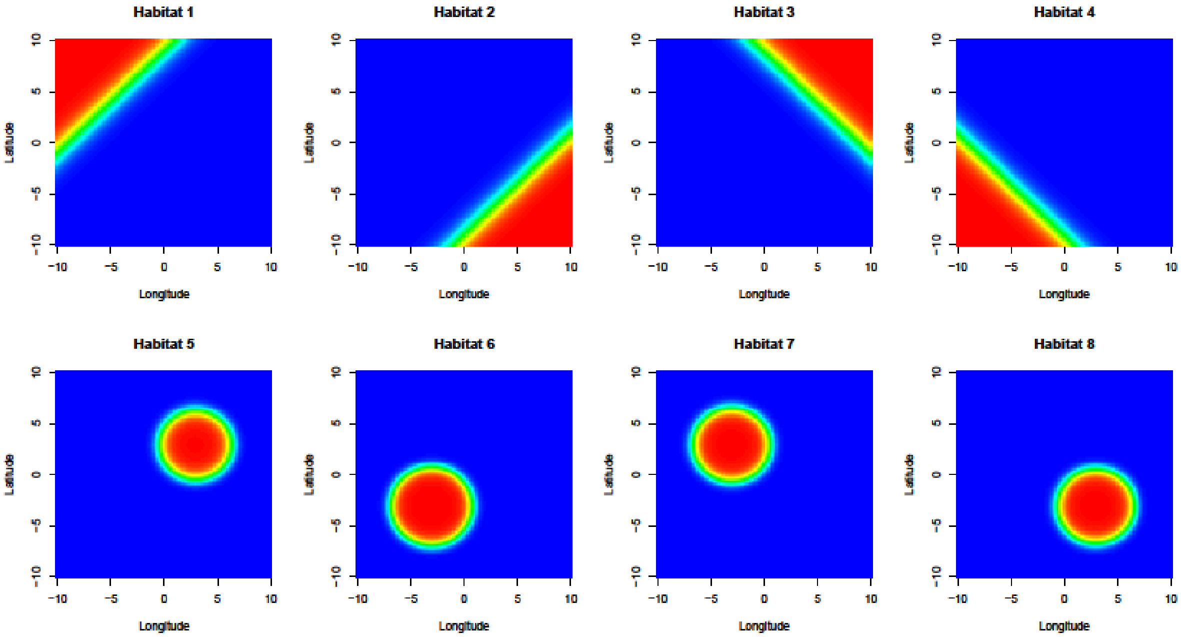

Figure 3.3.9: Individual logistic habitat probability plots for Scenario #7 (Diamond/Circles).

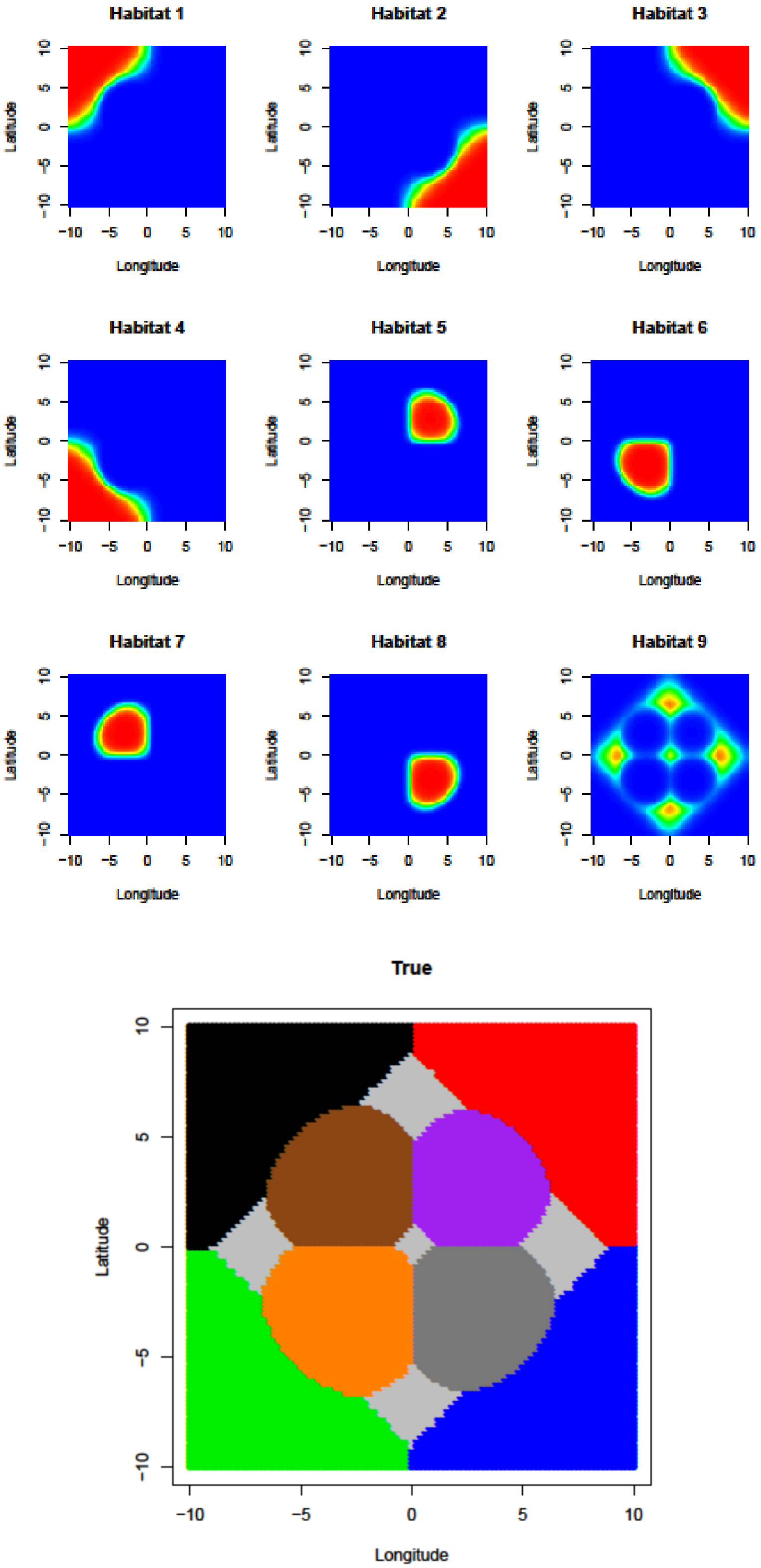

Figure 3.3.10: Additive logistic habitat probability plots (top) and habitat classification plot (bottom) for Scenario #7 (Diamond/Circles).

3.4 Summary of Scenarios

The next chapter will provide a lengthy analysis of simulation results for these 7 scenarios. Table 3.4.1 below summarizes the key feature of the scenarios, and may serve as a useful reference for the remainder of this paper.

Table 3.4.1: Summary of habitat scenarios.

Scenario	***H***	**n**	**S**	**Highest Polynomial Effect in X_1 and X_2**	**Other Investigations**
1) Baseline Linear	3	200	20	Linear	MC, Choice of n and S, etc.
2) Two Circles	3	150	15	Quadratic	MC, Boundary Width Analysis
3) Cubic	3	150	15	Cubic	MC, Convergence
4) Lin/Quad/Cube	4	225	24	Cubic	MC, Covariate Model Selection
5a) Diamond (Linear Model)	5	200	20	Linear	Model Misspecification
5b) Diamond (Quadratic Model)	5	200	20	Linear (Quadratic Model)	Model Misspecification
6) Four Circles	5	400	30	Quadratic	*H* Misspecification
7) Diamond/Circles	9	400	40	Quadratic	N/A

CHAPTER 4: RESULTS

4.1 Evaluation of Parameter Estimates

Assessing the success of estimation results for this model is multifaceted. Throughout the examples that follow I use a combination of three approaches for performance evaluation.

First, we can compare the true parameter values to the MLE parameter values directly. Two phenomena discussed in the next section, habitat swapping and steepness misspecification, however, make direct comparison difficult, particularly for the ***B*** parameter matrix. In some cases we can get a better indication of the estimation performance by considering the derived values π_{ih} and μ_{jh} instead.

A second method for assessing estimation performance is simply to compare estimated individual logistic habitat probability plots, additive logistic habitat probability plots, and habitat classification plots to their true counterparts. This is an indirect evaluation of estimation ***B***, whose values define the habitat plots, however it is possible—and relatively common—for the model to produce accurate habitat mappings while the parameter estimates, $\widehat{\boldsymbol{B}}$, are not accurate.

A final approach that is useful for assessing ***B*** estimation is to consider how many sites we classify correctly into habitats. To do this, I assign each sampled site to the most likely habitat (i.e. the habitat with the largest π_{ih} value at that site). Then I do the same thing for each site using the estimated probabilities, $\widehat{\boldsymbol{\pi}}$. To evaluate performance, I calculate the proportion of sites for which the classifications agree. Considering this classification success rate for estimation evaluation has the advantage of producing a single number to summarize performance. However this is artificial and potentially misleading in the same way that the classification plot is because sites are never actually attributed to a specific habitat. Nevertheless, this success rate is a useful summary which I use to present estimation results throughout Chapter 4.

4.2 Estimation Results

In the following subsections I describe the results from the seven scenarios used to test the model and estimation performance. The scenario specifications range from simple to complex and the estimation problems range from easy to challenging. Interestingly, some of the simplest scenarios are not necessarily the easiest to achieve good estimation performance.

As discussed in Section 3.1, simulation evaluation of estimation performance begins by specifying a true habitat scenario (in terms of B_{kh}, α_j and τ_{jh} parameters). The number of sites and species are important components of scenario specification. Section 4.3 will investigate the effect that different combinations of n and S have upon estimation. Here I provide results for a single combination of n and S for each scenario. Then a set of sites must be sampled before we may simulate data and run the estimation algorithm. Starting values for likelihood optimization must also be chosen (see Sections 3.2.6 and 4.2.8).

A more comprehensive comparison of results would control for random variation due to the choice of sites and starting values. In four of the seven scenarios I consider below, I conducted Monte Carlo replications, repeatedly sampling random site sets. These cases were: Scenario #1 (Baseline Linear), Scenario #2 (Two Circles), Scenario #3 (Cubic), and Scenario #4 (Linear/Quadratic/Cubic). For each of these examples, the results corresponding to the sample which provided the Monte Carlo median classification success rate are reported here. Detailed results of the Monte Carlo simulation may be found in Section 4.3.2. For the remaining three scenarios, Monte Carlo simulation was not feasible due to long computation times.

4.2.1 Baseline Linear Scenario

For the Baseline Linear scenario we use a sample of n=200 sites and S=20 species. Although these numbers may seem large, recall from Section 3.4 that realistic applications may entail more like n=1200 sites and S=200 species. The true and estimated $\boldsymbol{B}^T$ parameter matrices are:

$$\boldsymbol{B}^{\mathrm{T}} = \begin{bmatrix} 0 & 0.4 & -0.2 \\ 0 & 0 & 1 \end{bmatrix} \qquad \widehat{\boldsymbol{B}}^{\mathrm{T}} = \begin{bmatrix} 0.003 & 0.87 & -0.24 \\ 0.090 & 0.127 & 2.19 \end{bmatrix} \qquad where\ \boldsymbol{X}_i^T = (1 \quad X_1 \quad X_2)$$

Although these estimates seem poor, the model exhibits a feature suggestive of overparameterization or collinearity in the sense that many different $\widehat{\boldsymbol{B}}$ can produce reasonably good habitat probability estimates. Even for this relatively small parameter matrix, it is not easy to tell exactly how well our

Goodness of Fit Graphs

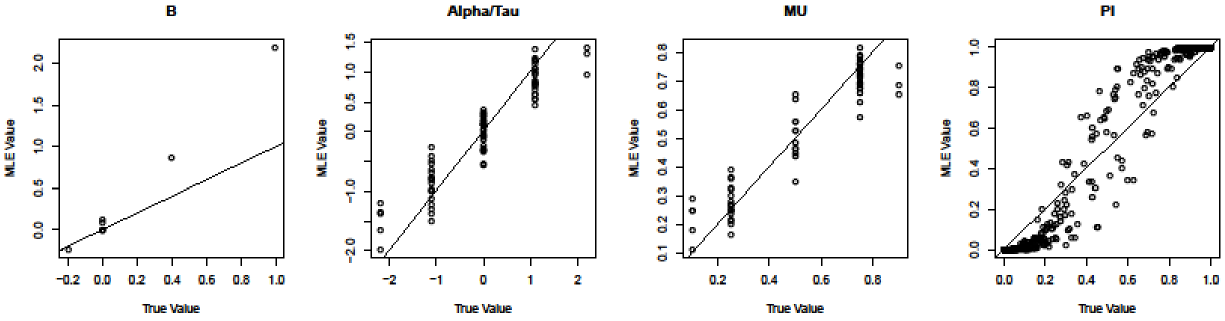

Figure 4.2.1: Comparison of true and MLE values of parameters and re-expressed parameters for Scenario #1 (Baseline Linear).

estimation has worked. This difficulty increases with larger ***B*** matrices. In this small example, the ***A*** parameter matrix (consisting of α_j and τ_{jh} values) has $S \times (H+1) = 80$ elements, making pointwise comparison of estimated parameter values difficult.

Figure 4.2.1 displays plots of MLE results against true values for the B, α, and τ parameters, as well as for the derived values μ and π. If the estimation were perfect we would find every point lying along the 45° line in each plot. The leftmost panel in Figure 4.2.1 shows that values in $\widehat{\boldsymbol{B}}$ generally have higher magnitude than their true counterparts in this estimation example. In particular, the points in this plot lie approximately along a line with a slope of 2. Because the estimates are about twice the magnitude of the true values in $\boldsymbol{B}$, we expect that the estimated habitats have boundaries that are approximately twice as sharp as those of the true habitats.

The second and third panels in Figure 4.2.1 show the estimation results for the α_j and τ_{jh} parameters and the corresponding μ_{jh} values respectively (recall that ***A*** is composed of α_j and τ_{jh} values as defined in Section 2.1). We find evidence that ***A*** is estimated well since estimates form a reasonably tight pattern around the true values. Recalling that $\mu_{jh} = \text{logit}(\alpha_j + \tau_{jh})$, it is not surprising that these plots look alike. Thus, in future examples I will not display both.

The rightmost panel in Figure 4.2.1 plots the estimated habitat membership probabilities, $\hat{\pi}_{ih}$, against their true values for each of the locations in the sample. There are 600 points shown in the plot representing the three $\hat{\pi}_{ih}$ values at each of the *n*=200 sites. Notice the 'S' shape of the points in the plot. We see that the $\hat{\pi}_{ih}$estimates are likely to underestimate the true values when that true π_{ih} is

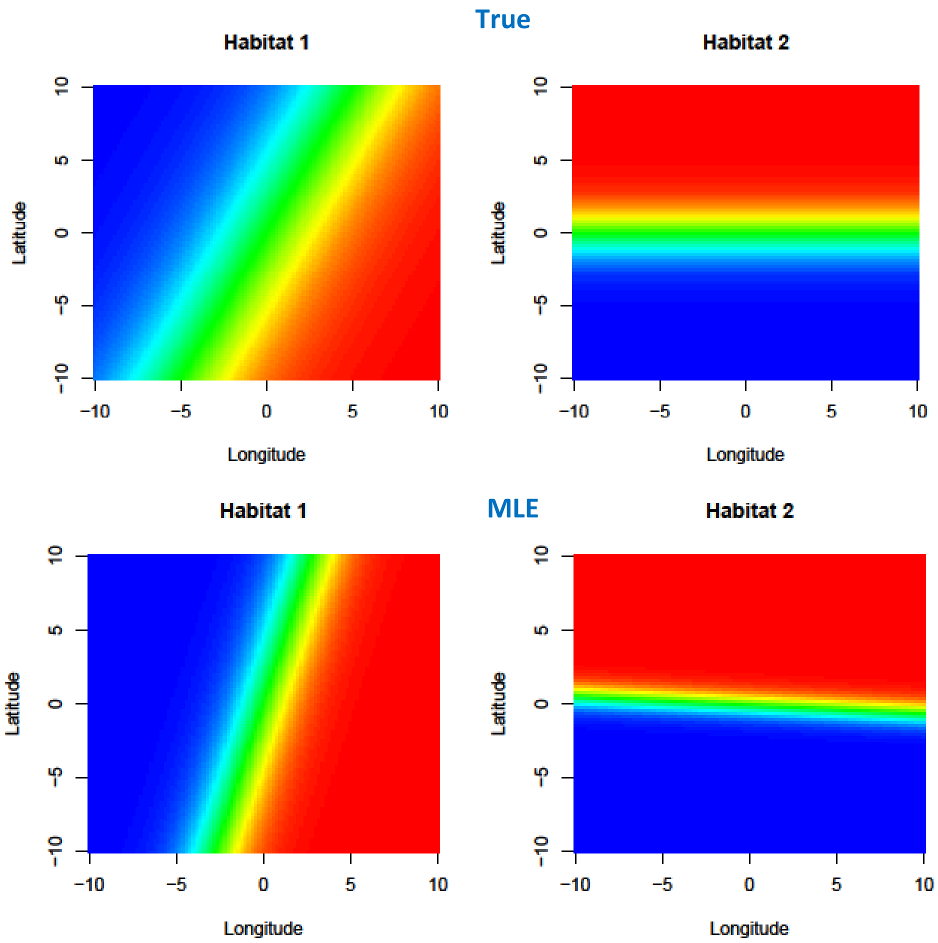

Figure 4.2.2: True (top) and estimated (bottom) individual logistic habitat probability plots for Scenario #1 (Baseline Linear).

below 0.5, and overestimate when the true π_{ih} is above 0.5. This feature tells the same story as the leftmost plot suggesting that estimated habitat boundaries are sharper than the true habitat boundaries.

Next I examine the habitat probability plots (Figure 4.2.2). The top two panels in this figure are the same two true individual logistic habitat probability maps for the linear baseline scenario that were presented in Section 3.3. The bottom two panels show the individual logistic habitat probability maps generated from $\widehat{\boldsymbol{B}}$. The estimated maps look very similar to the true maps with only a slight difference in the positions and orientations of each of the habitat boundaries. More noticeably, the estimated habitats have sharper boundaries. This agrees with the results shown in Figure 4.2.1, where we realized that the estimated values for the $\boldsymbol{B}$ parameter matrix looked more like a scalar multiple of $\boldsymbol{B}$. As predicted, the habitat boundaries are about twice as narrow in the estimated habitat maps.

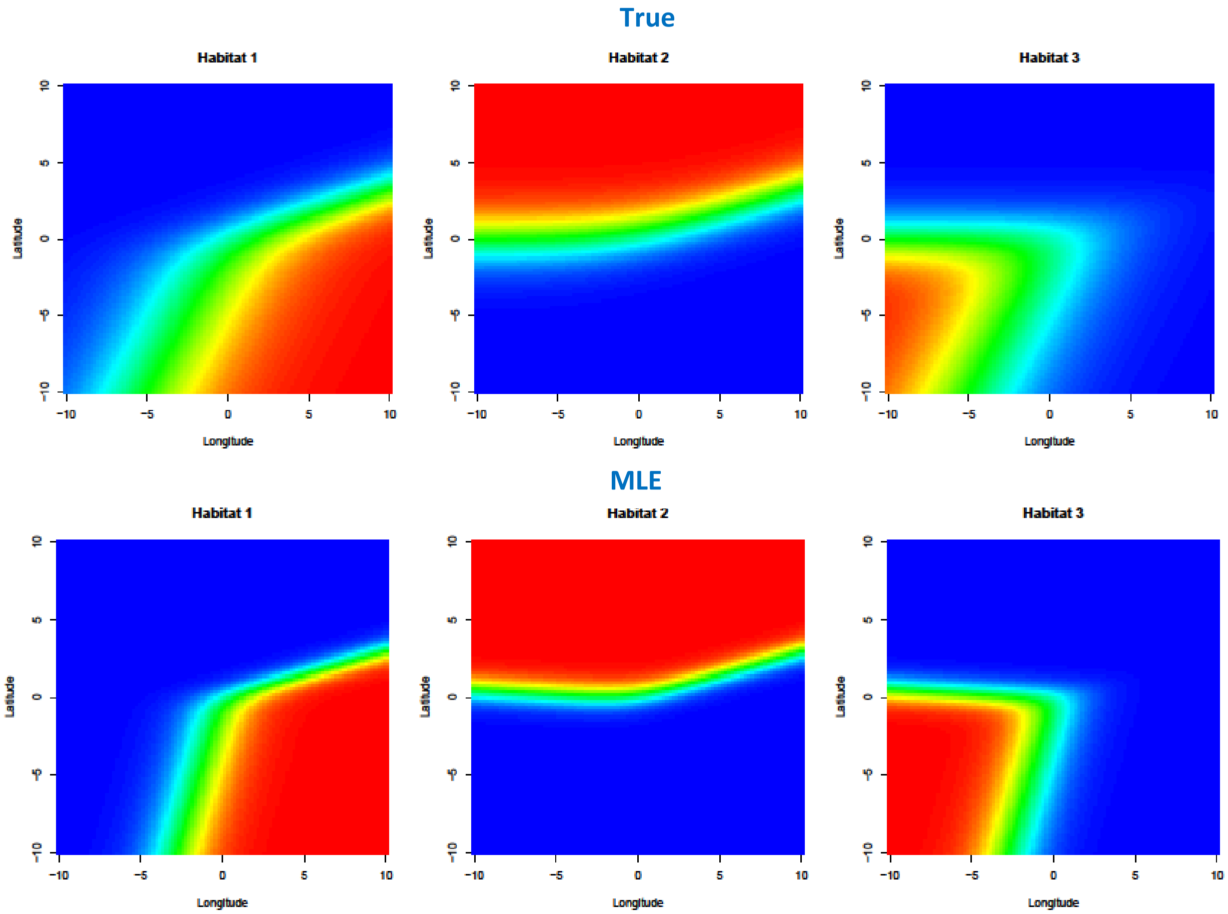

Figure 4.2.3: True (top) and estimated (bottom) additive logistic habitat probability plots for Scenario 1 (Baseline Linear).

As we did while initially considering the Linear Baseline scenario in Section 3.3, we may inspect the additive logistic habitat probability plots that are generated through the additive logistic model of the true and estimated *B* parameters. Here, we see that this model and associated estimation algorithm do a good job of reconstructing the habitat maps of this linear baseline scenario. As stated before, these additive logistic habitat probability plots are constructed as an aid in understanding how the three habitats interact with one another. For most of the examples ahead, I will omit these plots in the interest of saving space.

Figure 4.2.4 displays the true and estimated habitat classification plots. These plots illustrate that habitat estimation for this example has been highly successful. In the remainder of this chapter, comparing

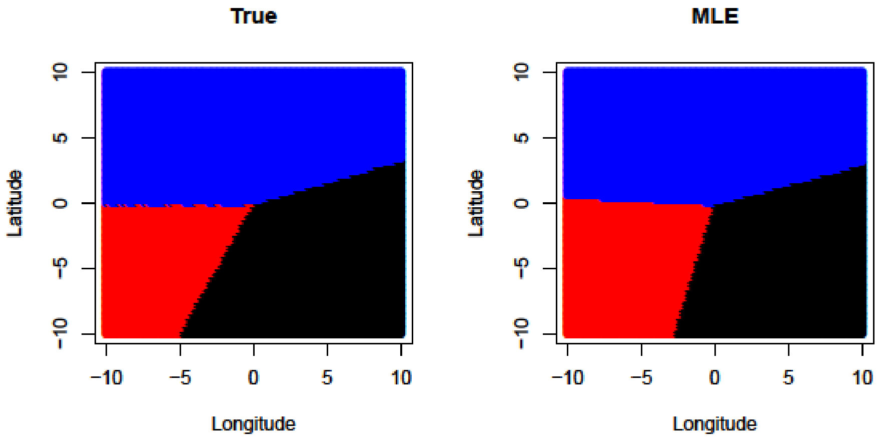

Figure 4.2.4: True (left) and estimated (right) habitat classification plots for Scenario #1 (Baseline Linear). Black=Habitat 1, Blue=Habitat 2, Red=Habitat 3.

classification plots between truth and MLE will be the most effective graphical diagnostic for assessing estimation performance. Hence, these plots will be included in every habitat estimation exercise in this paper.

The 'Classification Success Rate' Diagnostic Metric:

In Section 4.1, I introduced the idea of classifying each of the sampled sites into the most probable habitat. To do this I use the same method by which the classification plot is created, namely assigning each sampled site to the habitat for which the plurality of the additive logistic probability is attributed. Each classification decision reflects our best estimate of $\boldsymbol{z}$, the binary vector with all zeros except for a single 1 in the position corresponding to the habitat to which the given site belongs.

The leftmost panel in Figure 4.2.5 shows the habitat classifications for the 200 sites in the sample based on the $\boldsymbol{B}$ that specifies this scenario. The same color scheme that was used in the classification plot is applied here, and the resemblance is obvious. The only difference here is that I compute the estimated classification at each site in the sample rather than along a systematic grid that spans the study region. The right-hand plot in this figure shows the habitat classifications for the same sites based on $\hat{B}$. The two plots disagree on only seven habitat classifications, giving a classification success rate of 193/200=0.965. The plot is not necessary to compute the success rate, hence only the number is reported hereafter.

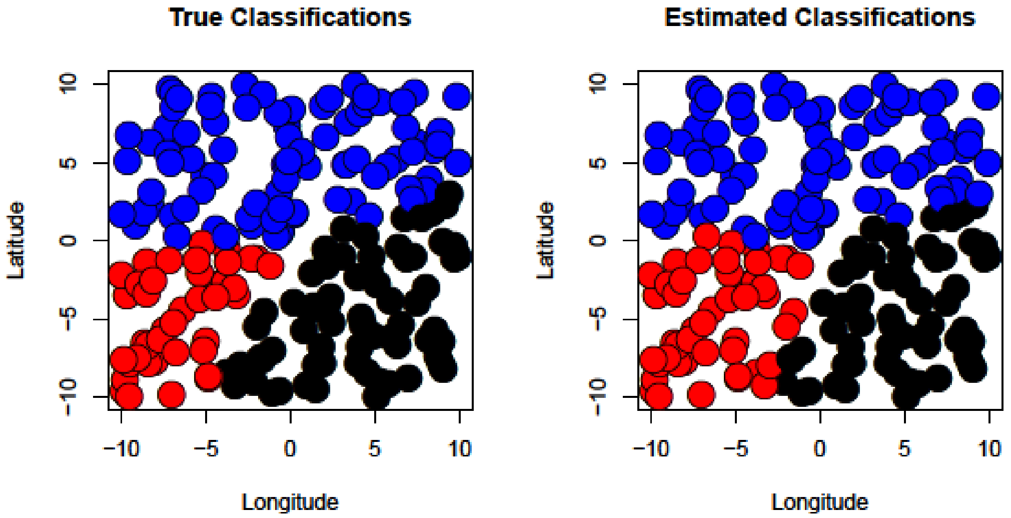

Figure 4.2.5: True (left) and estimated (right) sample habitat classification plots for Scenario #1 (Baseline Linear).

A Note on 'Habitat Swapping':

The estimation results presented above possessed the convenient feature that the colors assigned to the ML estimates matched the true color assignments. This is not guaranteed to be the case, and depends on the choice of parameter starting values for the algorithm and the number of habitats in the system.

To illustrate this, consider the following example. Another estimation attempt is made for this Baseline Linear scenario under a different set of parameter starting values[3]. The result is shown in Figure 4.2.6. Again, the individual logistic habitat probability maps are displayed above the estimated individual logistic habitat probability maps, but this time there appears to be little resemblance between the two pairs of maps.

Upon initial inspection of these plots, one might guess that this estimation is completely incorrect. However, a closer look will show that this is not the case. Figure 4.2.7 displays the true and MLE habitat classification plots, indicating that while the individual logistic habitat labels have changed locations, the general shape of the habitat boundaries looks reasonably accurate.

[3] For this example, $A_{0,jh}$ and $B_{0,kh}$ values are set as zero for all j=1…S, k=1…p, and h=1…H. More investigation of parameter starting value strategies is included in Section 4.3.8.

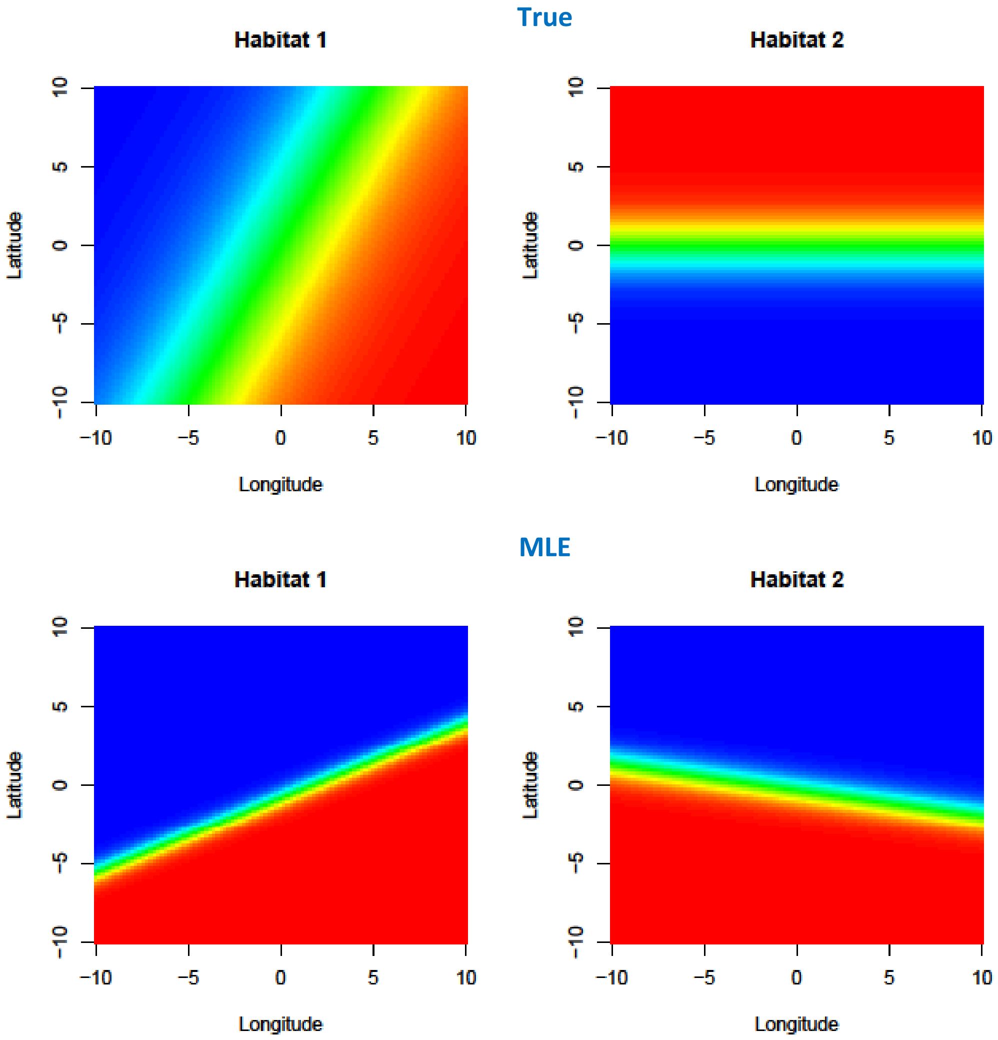

Figure 4.2.6: True (top) and estimated (bottom) individual logistic habitat probability plots for the example where colors were 'swapped' (see text).

Looking at these classification plots, we can see that the labels for Habitat 2 and Habitat 3 have been swapped in the estimated maps while Habitat 1 is located fairly accurately. We should not be surprised that this is a possible result of estimation, nor should we be perplexed when it occurs. The habitat names attached to these groups of locations with similar species presence/absence characteristics are completely artificial and arbitrary. In the context of simulation evaluation we may exchange these habitat labels freely after estimation, and in the context of a real world example the researchers will presumably want to denote each habitat with descriptive scientific names anyway. Despite this, it is necessary to 'properly' label the habitats in the context of simulation for us to be able to derive the classification success rate to aide in estimation evaluation and to ensure that the figures fairly represent the true performance.

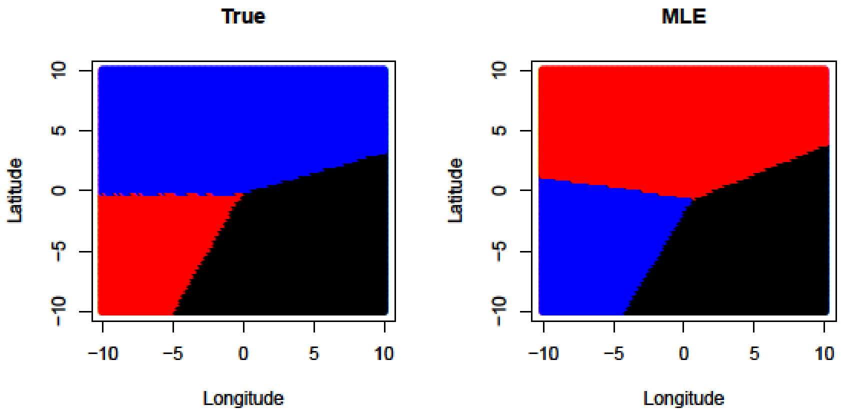

Figure 4.2.7: True (left) and estimated (right) habitat classification plots illustrating the swapping of Habitats 2 and 3 for Scenario #1 (Baseline Linear).

In this small example with very accurately estimated habitat boundaries, it is easy to visualize the switch necessary to put these habitats in their correct locations. However, if we need to estimate a higher number of habitats or decipher a set of less accurately estimated habitat boundaries, finding the proper habitat ordering may be considerably more difficult. For this reason, a clear definition is needed for determining the proper habitat labeling. I simply rearrange the labels/colors to be the arrangement that results in the highest classification success rate. From now on, this habitat swapping process will be treated as an internal component of the habitat estimation process for the purpose of performance evaluation. Results will be presented with habitats already swapped into their optimal permutation.

Figure 4.2.8 shows all the possible permutations of the three habitats alongside the true habitat classification plot. It is obvious that permutation #2 is the optimal order. Table 4.2.1 displays the classification success rates that correspond to each of the permutations shown in the plots. Note that the 0.96 success rate associated with swapping habitats 2 and 3 is much higher than any other ordering.

The last, implicitly defined habitat here has been estimated by the model directly, with explicit $\widehat{\boldsymbol{B}}$ parameters. This may become problematic in scenarios when the shape of the H^{th} habitat cannot conform to the modeled order. Moreover, it is not clear how to constrain B estimation to those first H-1 habitats when estimating habitat systems like Scenario #2, which we previously noted has a third habitat that cannot be explicitly defined by the quadratic framework that characterize the first two.

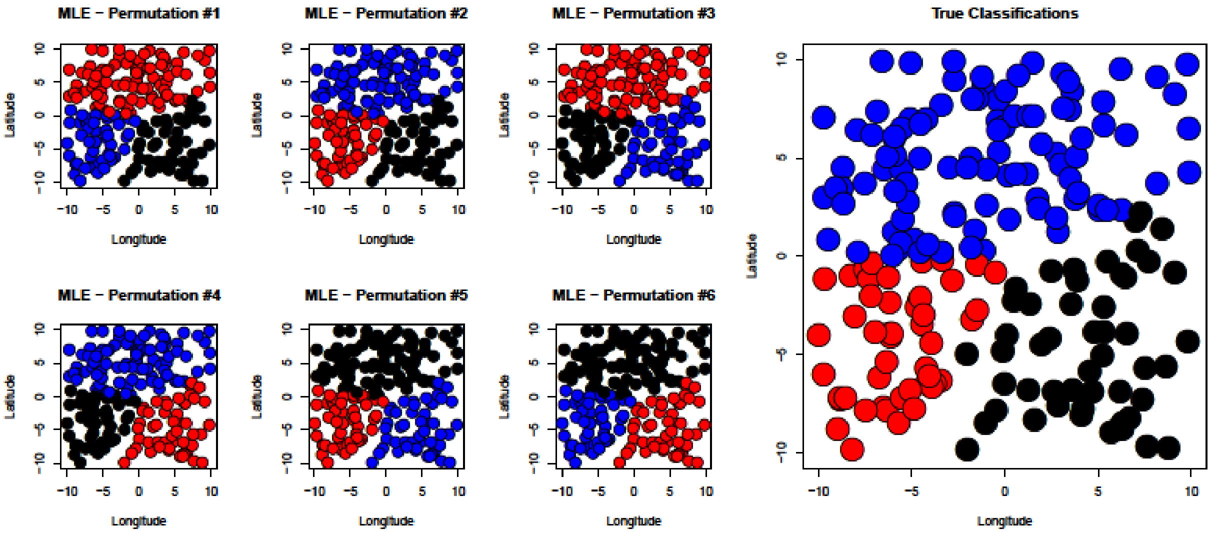

Figure 4.2.8: True (right) and estimated (left 6 panels) sample habitat classification plots with all possible habitat permutations for Scenario #1 (Baseline Linear).

Table 4.2.1: Success Rates for each of the possible permutations of estimated habitat labels. Permutation #2, having the highest classification success rate, is defined as the correct labeling.

Permutation #	1	2	3	4	5	6
Classification Success Rate	0.285	0.96	0.005	0.475	0.240	0.035

4.2.2 Two Circles Scenario

Recall that the simulation chosen to illustrate Scenario #1 (Two Circles) includes n=150 sites and S=15 species. The true and estimated $\boldsymbol{B}$ matrices are:

$$\boldsymbol{B}^{\mathrm{T}} = \begin{bmatrix} -12.75 & 5.10 & -0.51 & 5.10 & -0.51 \\ -12.75 & -5.10 & -0.51 & -5.10 & -0.51 \end{bmatrix}$$

$$\widehat{\boldsymbol{B}}^{\mathrm{T}} = \begin{bmatrix} -216.88 & 119.28 & -12.43 & 118.63 & -12.73 \\ -13.87 & -6.59 & -0.67 & -5.21 & -0.54 \end{bmatrix}$$

$$where \quad \boldsymbol{X}_i^{\mathrm{T}} = (1 \quad X_1 \quad X_1^2 \quad X_2 \quad X_2^2)$$

While Habitat 2 is estimated roughly correctly, the MLE parameters, $\widehat{\boldsymbol{B}}$, for Habitat 1 are approximately 20 times the magnitude of their true counterparts. Figure 4.2.9 displays true and estimated individual logistic habitat probability plots, and as we should expect, Habitat 1 estimate exhibits a much sharper boundary than the truth. Notwithstanding this, estimation of the habitat boundaries is very accurate. Moreover, this accuracy has been achieved with a smaller sample size and more parameters to estimate than in Scenario #1 (Baseline Linear). This is suggests that spatially distinct habitats, like the two circular habitat regions in this scenario, are easier to estimate than habitats which overlap like the upper right region in Scenario #1 where two habitats with individual membership probabilities near one compete. For this simulation-estimation exercise (Two Circles Scenario), we achieve a success rate of 0.98.

These results are impressive in terms of two of our three diagnostic methods, namely the classification success rate and the estimated habitat maps. Now let's consider the third diagnostic, a direct comparison of the true and estimated π_{ih} values. Though the π_{ih}'s are not directly estimated by the optimization algorithm, $\hat{\pi}_{ih}$'s are calculated from $\widehat{\boldsymbol{B}}$ and may be compared to the π_{ih}'s. The left-hand plot in Figure 4.2.10 graphs π_{ih}*'s* versus $\hat{\pi}_{ih}$'s. Most true π_{ih}'s are very near zero or one, and we see that these are estimated accurately. Among the sites that have intermediate probabilities, the 'S' shape found in this plot reflects the fact that Habitat 1 boundary probabilities are estimated to be overly abrupt. We would like to confirm or refute this suspicion by better understanding exactly which types of sites tend to be estimated poorly.

The right-hand panel in Figure 4.2.10 shows the distribution of estimation error among our π_{ih}'s. Only a few of the π_{ih} estimation errors deviate substantially from zero, but some that do err by as much 0.7. In the MLE classification plot of Figure 4.2.11, I overlay a set of yellow dots. The area of each dot is proportional to the squared error for $\hat{\pi}_{ih}$ at site *i* within habitat *h*. Note that the larger errors tend to fall along habitat boundary lines.

The conclusion here is clear: sites for which habitat allegiance is uncompetitive are easily estimated, while sites on or near a habitat boundary with two or more competitive candidate habitat assignments are more likely to be incorrectly classified.

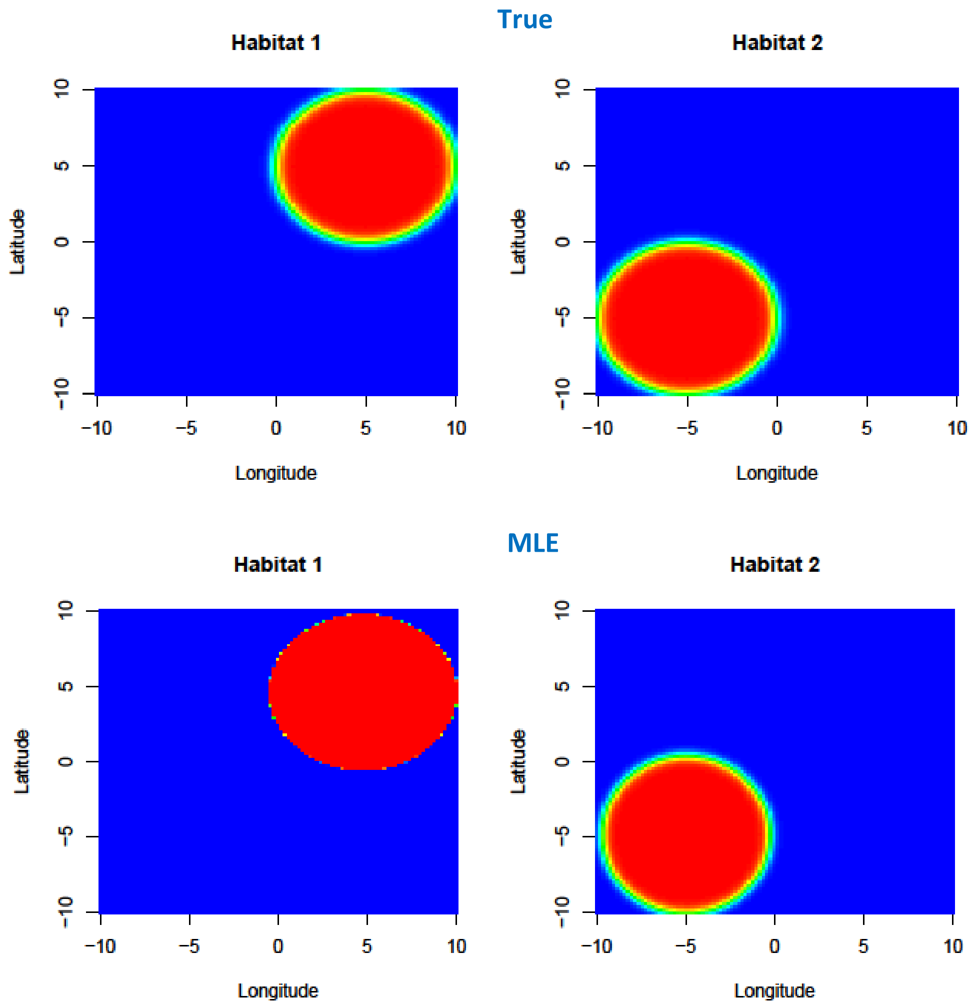

Figure 4.2.9: True (top) and estimated (bottom) individual logistic habitat probability plots for Scenario #2 (Two Circles).

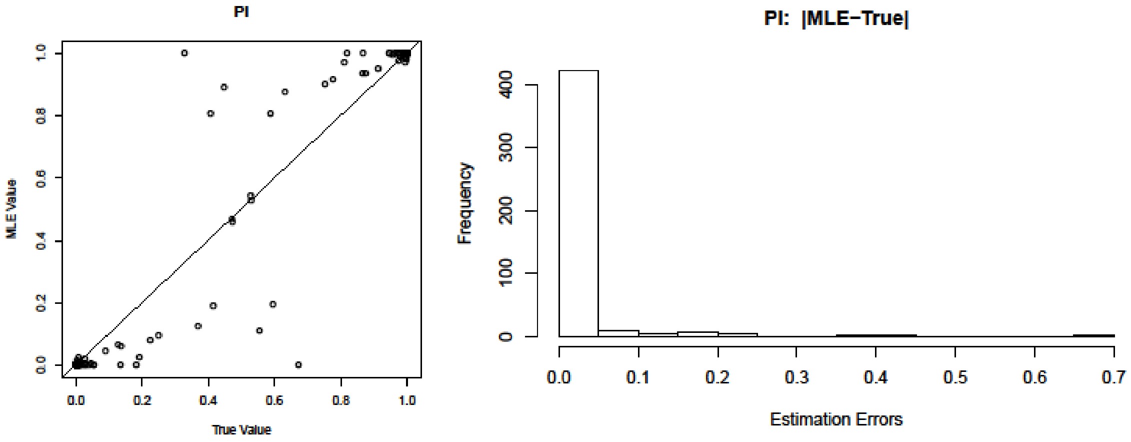

Figure 4.2.10: Scatter plot of $\boldsymbol{\pi}$ vs. $\hat{\boldsymbol{\pi}}$ (left) and histogram (right) of $\hat{\boldsymbol{\pi}}$ estimation errors for Scenario #2 (Two Circles).

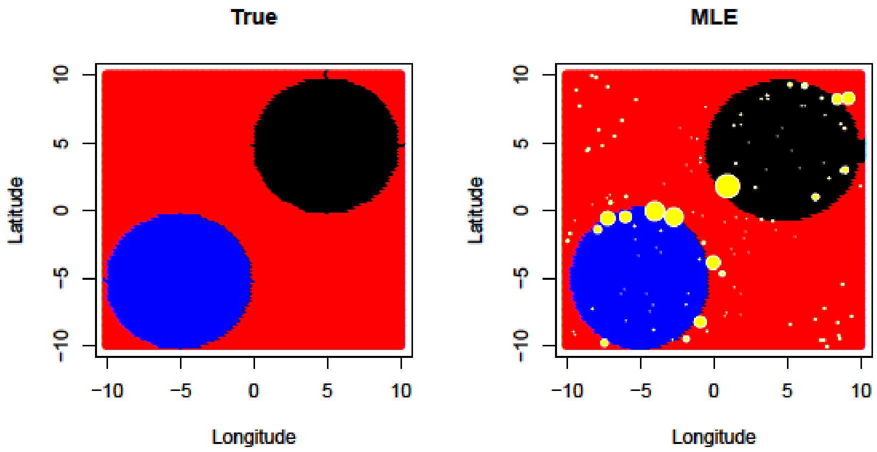

Figure 4.2.11: True (left) and estimated (right) habitat classification plots with yellow point area proportional to squared π-estimation error for Scenario #2 (Two Circles). Black=Habitat 1, Blue=Habitat 2, Red=Habitat 3.

4.2.3 Cubic Scenario

Recall from Section 3.3 that Scenario #3 (Cubic) employs *n*=150 sites and *S*=15 species, and has habitats characterized by the following matrices:

$$\boldsymbol{B}^{\mathrm{T}} = \begin{bmatrix} 10 & 10 & -0.4 & -0.2 & 40 & -0.07 & -0.7 \\ -15 & -10 & 0.4 & 0.2 & -2 & 0.3 & 0.08 \end{bmatrix}$$

$$and \quad \boldsymbol{X}_i^{\mathrm{T}} = (1 \quad X_1 \quad X_1^2 \quad X_2 \quad X_2^2)\,.$$

Estimation of ***B*** yields:

$$\widehat{\boldsymbol{B}}^{\mathrm{T}} = \begin{bmatrix} 30.19 & 12.98 & -0.16 & -0.32 & 50.90 & 0.11 & -0.81 \\ -21.22 & -21.48 & 1.04 & 0.39 & -4.22 & 0.48 & 0.20 \end{bmatrix}.$$

Below, I will show that poor estimation of ***B*** still yields excellent habitat estimation. Indeed the classification success rate is 0.973.

True and estimated individual logistic habitat probability maps are presented in Figure 4.2.12. In these results we find further evidence that the model in Foster, et al. (2011), in combination with the blocked non-linear Gauss-Seidel estimation algorithm described in Section 2.3, is an effective approach for relating species presence/absence and covariate data to the underlying habitat shape, location and characterizing species profile. At this point I have increasing confidence that this model will be able to fit more complex scenarios including more covariates, spatially correlated covariates, and more habitats.

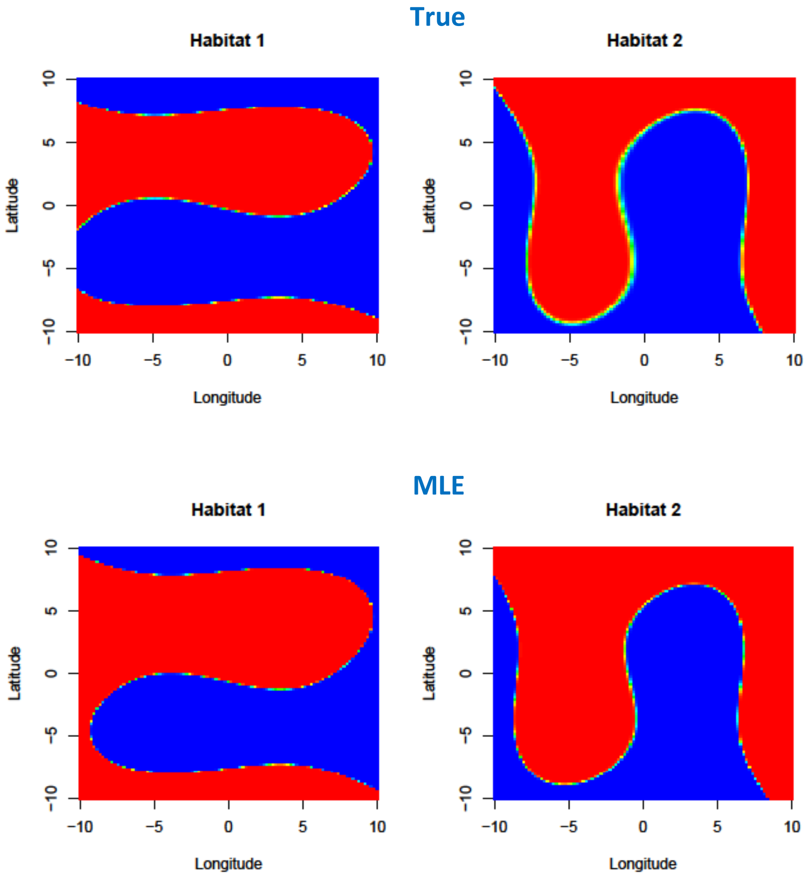

Figure 4.2.12: True (top) and estimated (bottom) individual logistic habitat probability plots for Scenario #3 (Cubic).

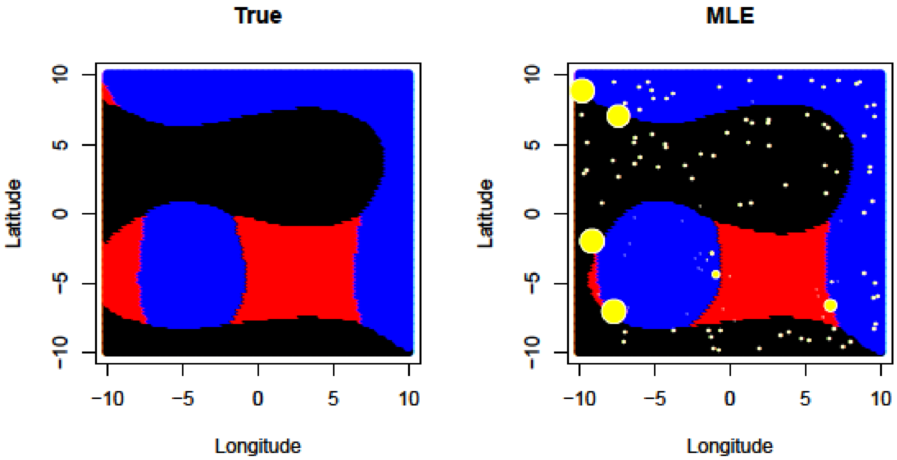

Figure 4.2.13: True (left) and estimated (right) habitat classification plots with yellow point area proportional to squared π-estimation error for Scenario #3 (Cubic). Black=Habitat 1, Blue=Habitat 2, Red=Habitat 3.

The true and estimated habitat classification plots for Scenario #3 are shown (Figure 4.2.13) with yellow dots indicating $\boldsymbol{\pi}$ estimation error as in the previous scenario. In Section 3.3 I proposed a possible ecological interpretation of this scenario as including prime (black), fringe (red) and non-habitat (blue) regions. Using this idea, the prime habitat represents ideal conditions for a set of species whose presence characterize a habitat, the fringe habitat region represents the area for which those species have moderately favorable conditions and the non-habitat region is area for which presence of those species is not favored by the local covariates. In this case, the estimated habitat classification plot above shows that the boundaries between these habitat categories are estimated fairly accurately. Unsurprisingly, higher estimation error occurs along the two smaller fringe habitat portions on the left side of the map where the three habitat categories are in close proximity (competition).

4.2.4 Linear/Quadratic/Cubic Scenario

This scenario (see Section 3.3) is the first to include four habitats. Because of its increased complexity, it employs a larger sample of *n*=225 sites and *S*=24 species. The specifications are:

$$\boldsymbol{B}^{\mathrm{T}} = \begin{bmatrix} -14.85 & -1.98 & 0 & 0 & -0.40 & 0 & 0 \\ 4.46 & -0.30 & -2.38 & 0.18 & 0.59 & -1.19 & 0.24 \\ -34.20 & 7.63 & -0.55 & 0 & -7.63 & -0.55 & 0 \end{bmatrix}$$

$$and \qquad \boldsymbol{X}_i^{\mathrm{T}} = (1 \quad X_1 \quad X_1^2 \quad X_1^3 \quad X_2 \quad X_2^2 \quad X_2^3)$$

The estimated parameters are:

$$\widehat{\boldsymbol{B}}^{\mathrm{T}} = \begin{bmatrix} -108.02 & 5.01 & 0.47 & -0.11 & -19.70 & -0.35 & 0.17 \\ 542.54 & -14.37 & -337.77 & 22.41 & -196.31 & -62.53 & 24.80 \\ -700.56 & 108.72 & 2.55 & -1.00 & -192.00 & -17.18 & -0.24 \end{bmatrix}$$

This estimation exercise achieves a success rate of 0.92.

The true and estimated individual logistic habitat probability maps are shown in Figure 4.2.14. While the shape and location of Habitats 2 and 3 are estimated accurately, the estimate for Habitat 1 underestimates the true size of the habitat. This error accounts for the 0.92 classification success rate, which is a smaller than we have found in previous examples.

It is obvious from the estimated individual logistic habitat probability maps and that once again estimated habitat boundaries are much too abrupt. Throughout my experience of fitting many simulated scenarios

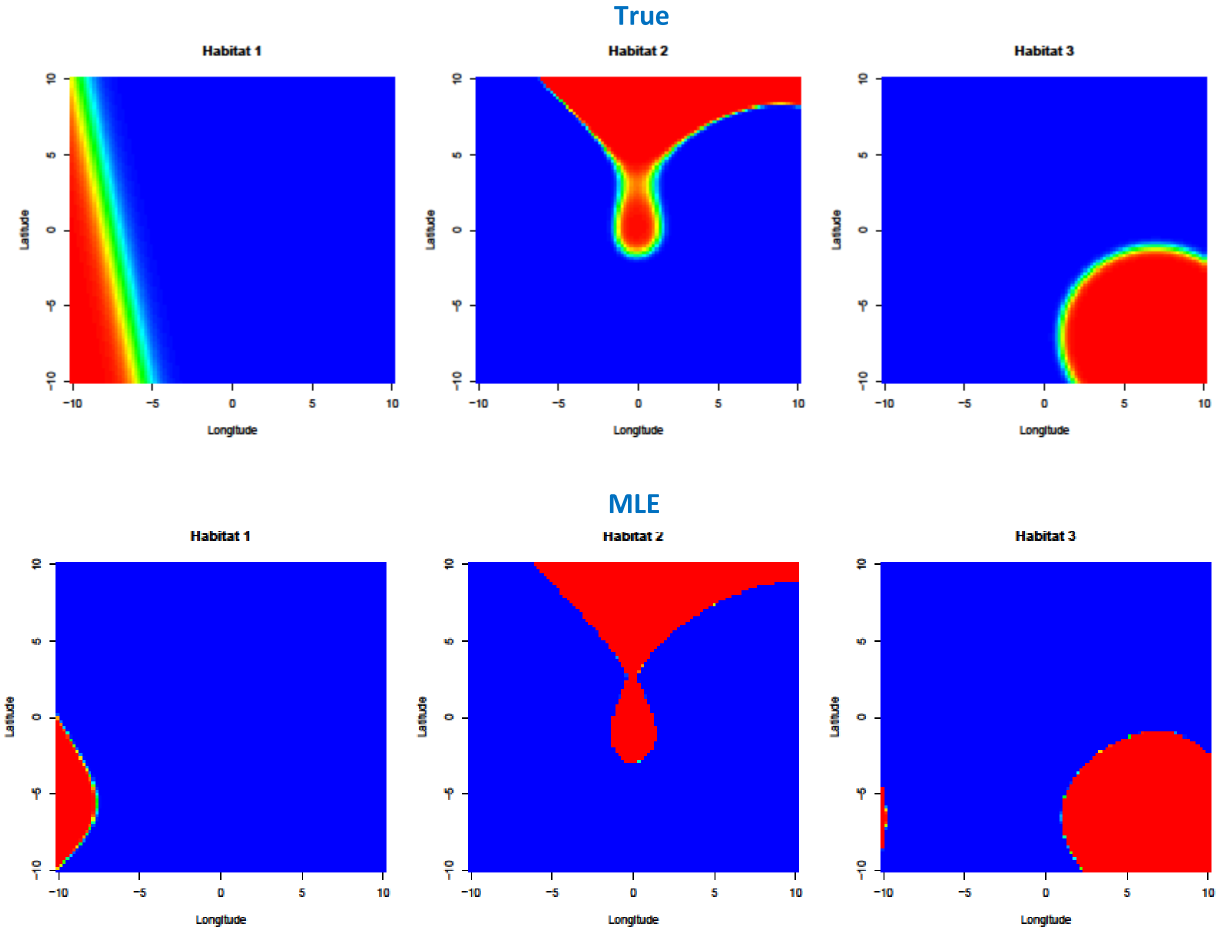

Figure 4.2.14: True (top) and estimated (bottom) individual logistic habitat probability plots for Scenario #4 (Linear/Quadratic/Cubic).

with this model, it has been much more common for this error to occur than for estimated boundaries to be insufficiently sharp. I believe that this problem might be mitigated if I inserted either 1 or *H*-1 parameters to scale modeled probabilistic habitat boundaries. A single scaling parameter would scale all the boundaries equally, whereas the larger set of parameters would scale each habitat's probabilistic boundaries individually. However, it is not currently clear to me how to do so in a way that ensures identifiability. This topic is discussed further in Section 4.3.9.

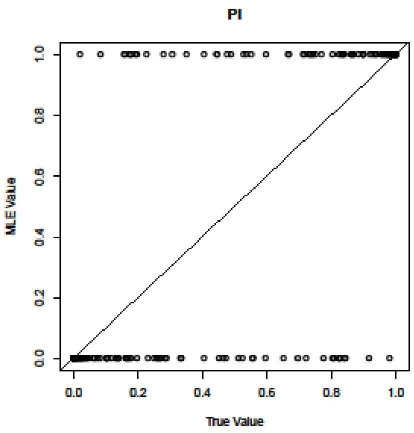

Figure 4.2.15: Scatter plot of $\boldsymbol{\pi}$ vs. $\hat{\boldsymbol{\pi}}$ for Scenario #4 (Linear/Quadratic/Cubic).

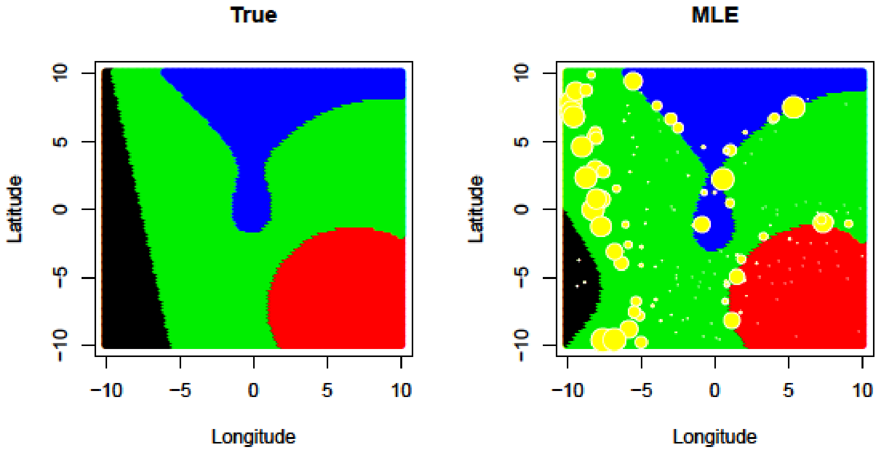

Figure 4.2.16: True (left) and estimated (right) habitat classification plots with yellow point radius proportional to estimation error for $\boldsymbol{\pi}$ for Scenario #4 (Linear/Quadratic/Cubic). Black=Habitat 1, Blue=Habitat 2, Red=Habitat 3, Green=Habitat 4.

Figures 4.2.15 and 4.2.16 examine the estimation errors. The scatter plot in Figure 4.2.15 shows that despite the existence of intermediate true π probabilities, all estimates π values are near zero or one. This is evidence for the most severe steepness misestimation presented so far. In the right panel of Figure 4.2.16 we find errors along habitat boundaries again. Unsurprisingly, large errors also exist where Habitat 1 (black) is misclassified as Habitat 4 (green).

4.2.5 Diamond Scenario

This scenario was introduced in Section 3.3. In addition to evaluating estimation performance I will investigate the effect of mis-specifying the model form. Specifically I will present results when a linear truth is fit with a quadratic polynomial in the covariates. Results of fitting a model of the correct form will also be provided for comparison. First, the results of the correct model specification (which I will refer to as Scenario #5a) are shown below. Here, $\boldsymbol{B}_1^T$ is the same parameter matrix that is displayed in Section 3.3, and $\widehat{\boldsymbol{B}}_1^{\mathrm{T}}$ is the linear fit.

$$\boldsymbol{B}_1^T = \begin{bmatrix} -10.69 & -1.19 & 1.19 \\ -10.69 & 1.19 & -1.19 \\ -10.69 & 1.19 & 1.19 \\ -10.69 & -1.19 & -1.19 \end{bmatrix} \qquad \widehat{\boldsymbol{B}}_1^{\mathrm{T}} = \begin{bmatrix} -23.50 & -2.85 & 2.01 \\ -4399.44 & 478.78 & -439.64 \\ -1335.42 & 150.44 & 125.84 \\ -622.81 & -72.23 & -78.66 \end{bmatrix}$$

$$where \qquad \boldsymbol{X}_i^{\mathrm{T}} = (1 \quad X_1 \quad X_2)$$

Below, $\boldsymbol{B}_2^{\mathrm{T}}$ also represents the true scenario, where two columns of zeroes reflect the absence of true quadratic effects in a parameterization that would allow quadratic forms. Thus, $\boldsymbol{B}_2$ generates the exact same habitat maps as the one above. This parameterization will be referred to later as Scenario #5b. $\widehat{\boldsymbol{B}}_2^{\mathrm{T}}$ is an estimate of $\boldsymbol{B}_2^{\mathrm{T}}$ in the usual sense. Estimating $\boldsymbol{B}_2^{\mathrm{T}}$ can be seen as a method for overfitting $\boldsymbol{B}_1^T$.

$$\boldsymbol{B}_2^{\mathrm{T}} = \begin{bmatrix} -10.69 & -1.19 & 0 & 1.19 & 0 \\ -10.69 & 1.19 & 0 & -1.19 & 0 \\ -10.69 & 1.19 & 0 & 1.19 & 0 \\ -10.69 & -1.19 & 0 & -1.19 & 0 \end{bmatrix}$$

$$\widehat{\boldsymbol{B}}_2^{\mathrm{T}} = \begin{bmatrix} -697.35 & -21.06 & 3.96 & 123.86 & -4.40 \\ -1397.04 & 215.31 & -7.62 & -76.18 & 6.07 \\ -121.94 & 23.20 & -1.00 & 3.69 & 1.62 \\ -4410.14 & -670.00 & -21.16 & -592.53 & -16.87 \end{bmatrix}$$

$$with \qquad \boldsymbol{X}_i^{\mathrm{T}} = (1 \quad X_1 \quad X_1^2 \quad X_2 \quad X_2^2)$$

Figure 4.2.17 includes individual logistic habitat probability maps for the true parameters and estimates, with the two sets of MLE estimates being derived from the correct (linear) predictor and the incorrect (quadratic) predictor. The linear habitat estimates are accurate except for the familiar boundary sharpness problem. For the quadratic estimates, Habitat 1 is of particular interest. It is clearly not a good fit if estimating Habitat 1 is our only concern. However, when considering the additive logistic habitat system, Habitat 3 correctly dominates the upper right portion of the study region over Habitat 1. Evidence for this can be found in the classification plots in Figure 4.2.18. Thus, the overall estimation of the system of habitat maps is very good.

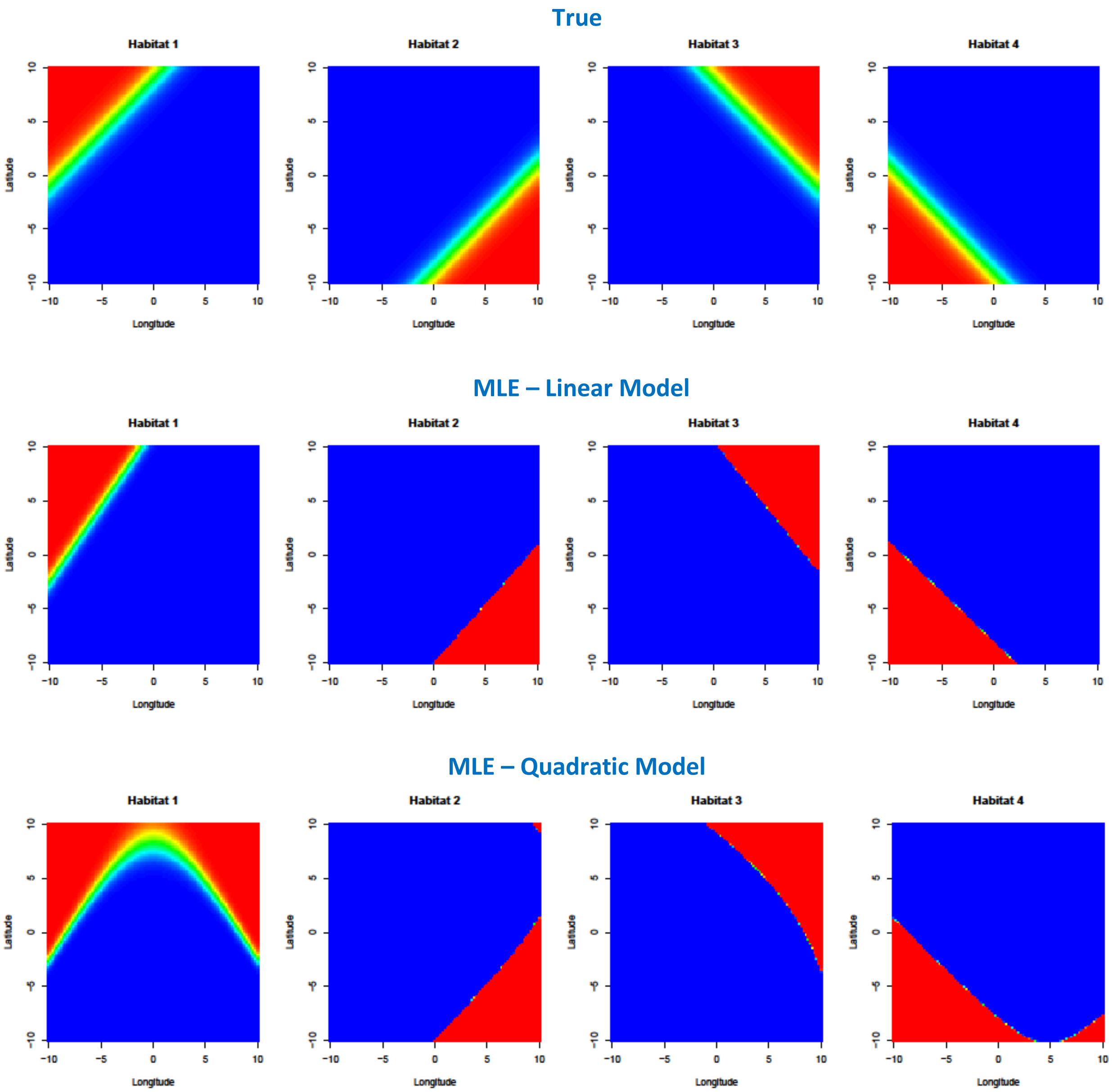

Figure 4.2.17: True (top), estimated with a linear model form (middle) and estimated with a quadratic model form (bottom) individual logistic habitat probability plots for Scenario #5 (Diamond).

The result for Habitat 1 suggests that the data in the upper right portion of the study region must include an unusually large proportion of sites and species presences that are similar to those found in Habitat 1. This feature results in confusion when estimating only the probability map for Habitat 1. However, the bottom panels in Figure 4.2.18 indicate that Habitat 3 dominates the top right region, assuring that the additive logistic habitat probability plots estimate the truth well.

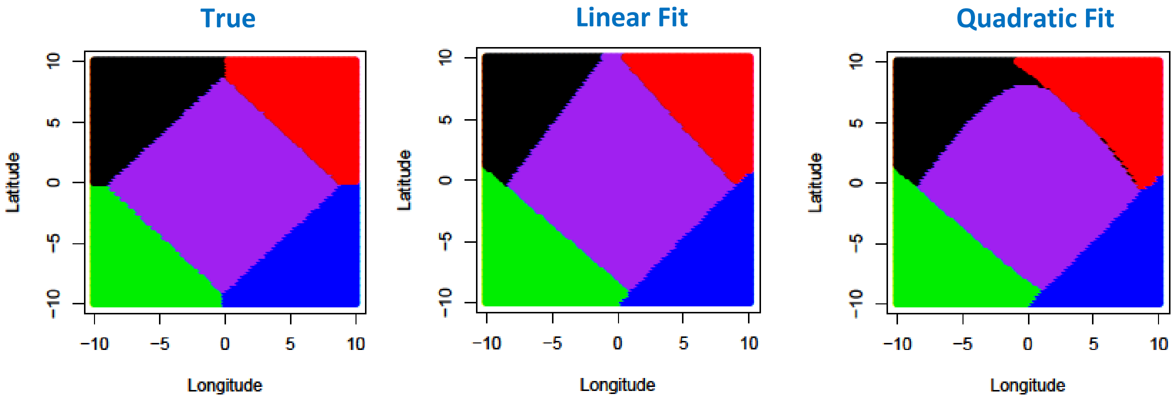

Figure 4.2.18: True (left), estimated with a linear model form (middle) and estimated with a quadratic model form (right) habitat classification plots for Scenario #5 (Diamond). Black=Habitat 1, Blue=Habitat 2, Red=Habitat 3, Green=Habitat 4, Purple=Habitat 5.

Viewing the true habitat classification plot next to the estimated versions corresponding to the linear and quadratic polynomial models, we see that both models can yield excellent estimated habitat maps. Moreover, model success rates are virtually identical: the linear model achieves a success rate of 0.945, while the quadratic model has a 0.95 success rate. This example raises questions about model selection that will be investigated in Section 4.3.5.

4.2.6 Four Circles Scenario

Recall from Section 3.3 that Scenario #6 (Four Circles) employs *n*=400 sites and *S*=30 species. The model is parameterized with:

$$\boldsymbol{B}^T = \begin{bmatrix} -12.75 & 5.10 & -0.51 & 5.10 & -0.51 \\ -12.75 & -5.10 & -0.51 & -5.10 & -0.51 \\ -12.75 & -5.10 & -5.10 & 5.10 & -5.10 \\ -12.75 & 5.10 & -5.10 & -5.10 & -5.10 \end{bmatrix} \qquad and \quad \boldsymbol{X}_i^{\mathrm{T}} = (1 \quad X_1 \quad X_1^2 \quad X_2 \quad X_2^2)$$

The estimates are:

$$\widehat{\boldsymbol{B}}^{\mathrm{T}} = \begin{bmatrix} -12677.36 & 5009.64 & -512.48 & 5611.76 & -571.24 \\ -84.24 & -37.53 & -3.80 & -30.97 & -3.16 \\ -2667.81 & -1062.17 & -107.26 & 1152.03 & -116.12 \\ -1060.58 & 462.11 & -48.15 & -506.02 & -53.63 \end{bmatrix}$$

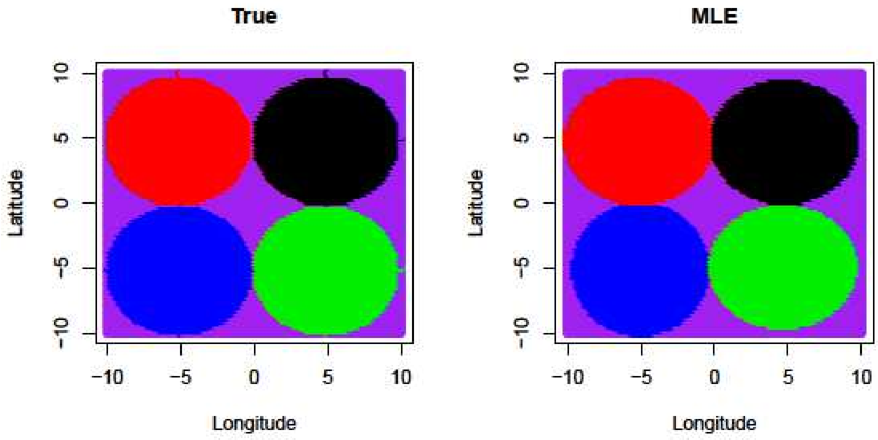

Figure 4.2.19: True (left) and estimated (right) habitat classification plots for Scenario #6 (Four Circles). Black=Habitat 1, Blue=Habitat 2, Red=Habitat 3, Green=Habitat 4, Purple=Habitat 5.

For 96.5% of the sampled sites, the true and estimated sample habitat classifications agree. The true and MLE classification plots are displayed in Figure 4.2.19. Here, we see another example of excellent shape and location estimation for a relatively complex 5-habitat scenario.

4.2.7 Diamond/Circles Scenario

This scenario uses simulated covariate data and species presence/absence information from *n*=400 sites and *S*=30 species to estimate the following model.

$$\boldsymbol{B}^{T}=\begin{bmatrix}-10.69 & -1.19 & 0 & 1.19 & 0\\ -10.69 & 1.19 & 0 & -1.19 & 0\\ -10.69 & 1.19 & 0 & 1.19 & 0\\ -10.69 & -1.19 & 0 & -1.19 & 0\\ -2.84 & 2.57 & -0.43 & 2.57 & -0.43\\ -1.44 & -2.57 & -0.43 & -2.57 & -0.43\\ -2.14 & -2.57 & -0.43 & 2.57 & -0.43\\ -2.14 & 2.57 & -0.43 & -2.57 & -0.43\end{bmatrix} \quad and \quad \boldsymbol{X}_i^{\mathrm{T}}=(1 \quad X_1 \quad X_1^2 \quad X_2 \quad X_2^2)$$

Estimation yields the following $\widehat{\boldsymbol{B}}$ and has a 0.932 classification success rate.

$$\widehat{\boldsymbol{B}}^{\mathrm{T}}=\begin{bmatrix}-845.44 & -140.33 & -6.64 & 47.14 & 4.82\\ -2191.99 & 318.57 & -9.44 & -144.59 & 7.59\\ -17969.64 & 1815.46 & 10.62 & 1626.66 & 45.71\\ -1176.32 & -182.10 & -6.93 & -119.79 & -1.11\\ -163.18 & 165.44 & -27.32 & 103.12 & -20.23\\ -11.21 & -21.80 & -3.69 & -16.55 & -3.03\\ -12.92 & -29.17 & -5.15 & 31.72 & -5.69\\ -188.69 & 153.34 & -23.18 & -104.13 & -16.72\end{bmatrix}$$

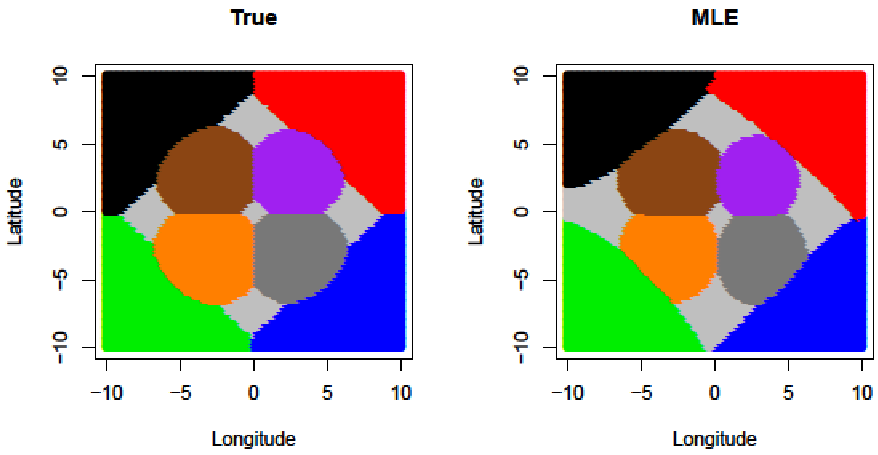

Figure 4.2.20: True (left) and estimated (right) habitat classification plots for Scenario #7 (Diamond/Circles). Black=Habitat 1, Blue=Habitat 2, Red=Habitat 3, Green=Habitat 4, Purple=Habitat 5, Orange=Habitat 6, Brown=Habitat 7, Dark Grey=Habitat 8, Light Grey=Habitat 9.

True and estimated habitat classification plots are presented in Figure 4.2.20. Using the statistical model of Foster, et al. (2011), and the estimation algorithm shown in Section 2.3 of this paper, we are able to estimate the 400 B_{kh}, α_j and τ_{jh} parameters that define this system of habitats. Despite the usual probability surface steepness misestimation apparent in $\widehat{\boldsymbol{B}}^{\mathrm{T}}$, the MLE habitat classification plot in Figure 4.2.20 is an excellent estimate.

4.2.8 Overview of Estimation Results

For the reader's convenience, a table summarizing the estimation results presented so far is included in this section. The MC Median column specifies whether, for a given scenario, the estimation results presented correspond to the random sample led to the Monte Carlo median classification success rate. For the scenarios in which a Monte Carlo simulation was not completed, the random seed used to generate the random sample of sites was chosen arbitrarily. Note that *n* and *S* values were chosen in part to achieve a relatively similar success rate across scenarios and are smaller than the values that might be used for the Great Barrier Reef application.

Table 4.2.2: Summary of estimation results for the seven scenarios.

Scenario	**MC Median**	**n**	**S**	**Success Rate**
1) Baseline Linear	Yes	200	20	0.965
2) Two Circles	Yes	150	15	0.980
3) Cubic	Yes	150	15	0.973
4) Lin/Quad/Cube	Yes	225	24	0.920
5a) Diamond (Linear Model)	No	200	20	0.945
5b) Diamond (Quadratic Model)	No	200	20	0.950
6) Four Circles	No	400	30	0.965
7) Diamond/Circles	No	400	40	0.932

Table 4.3.1: Computation times for the 7 scenarios with stated *n*, *S* combination listed in Table 4.2.2

Scenario	1	2	3	4	5(a)	5(b)	6	7
Time	11m 25s	7m 25s	8m 35s	28m 50s	28m 40s	35m 15s	2h 53m 48s	15h 8m 38s

4.3 Special Topics

Having presented basic estimation results in the previous sections, I now address specific topics relevant to the construction of habitat scenarios, simulation of data, and estimation of the model parameters.

4.3.1 A Note on Computation Time

Throughout the simulation and estimation phase of my testing of the model, management of computation time has been a constant concern. Some of the largest scenarios I ran (in terms of the number of sites, species, habitats, algorithm iterations and estimation method) required several days to complete estimation.

Simulations were run on two computers. The slower computer had a 2.8 GHz Pentium 4 processor with 1GB of RAM running Windows XP and 32-bit R v 2.12. The faster computer had a 2.4 GHz Intel® Core™ i5 CPU with 4GB RAM, running Windows 7 and a 64-bit version of R 2.12. The times listed in Table 4.3.1 and throughout this section will refer to computation time on the slower Windows XP machine. The faster Windows 7 machine takes only approximately 40% the time of the slower computer for a given simulation.

Some general rules of thumb were found:

- Computation time increases approximately linearly with both *n* and *S*, all else being equal.
- For large values of *S* (i.e. $S > 50$), computation time begins to grow roughly exponentially with *S*.
- Growth in the number of habitats, *H*, has a very large increasing effect on computation time.

4.3.2 Monte Carlo Simulation & The Effect of *n* and *S*

During the early testing of the estimation of this model, I noticed that simulation trials with identical assumptions produced very different results. Specifically, taking a different random sample of sites sometimes led to important differences in the estimated habitat maps. Based on this observation I speculate that the variability that exists within the sample of sites and simulation of data is more influential than I had previously expected. To address this inquiry, and to understand the model's variability better, I used additional Monte Carlo replication.

A Monte Carlo simulation study was performed on the random sample of site locations for the first four habitat scenarios. In each case, 50 replicated simulations were run, using the same scenario parameters (***B***) but different, random sites and covariate values. Additionally each of these Monte Carlo studies was repeated for various combinations of *n* and *S*. The details are as follows. For Scenario #1 (Baseline Linear), I compared all pairwise combinations of three values for each *n* and *S*. For Scenarios #2-#4 I use only two values for each *n* and *S*. Only 50 replications were used in each study in order to limit total simulation time.

For each MC trial, three measurements were taken and are presented in Tables 4.3.2: the median classification success rate (Rate), the Monte Carlo standard error of the 50 classification success rates (S.E.), and the median computation time (Time) in seconds among the 50 estimation tasks. We should expect an increase in median success rate and computation time as *n* increases, while the standard error of classification success rates should decrease when *n* grows. These trends are found to be true in the MC results for the Baseline Linear scenario. Additionally, we find that increasing *S*, controlling for *n*, has a very similar effect upon the three measures we are considering. The Monte Carlo standard error among all trials ranged from 0.015 to 0.142, the upper end of which is quite high and reflects the variable results mentioned above. Even though these high standard error values correspond to trials with especially low sample sizes, the topic of how estimation variation depends on site sample variation should be addressed further in future study and in application to real data.

In general, increasing *n* is the most reliable way to improve estimation results (recall that increasing *n* does not increase the number of parameters, while increasing *S* does). However Tables 4.3.2 show that increasing *S* has a very similar effect in most cases. Section 4.3.3 will investigate balance between *n* and *S* more directly.

Tables 4.3.2: Monte Carlo simulation results for Scenarios 1-4, and for various combinations of *n* and *S*. Here 'Rate' refers to the median success rate, 'S.E.' refers to the MC standard error, and 'Time' is the median computation time in seconds on the Windows XP machine described above.

Scenario #1: Baseline Linear

Monte Carlo Simulation		*# of Species, S*		
		5	**10**	**20**

# of Sites, n	**50**	Rate = 0.88 	
		S.E. = 0.079
		Time = 55
			Rate = 0.90
			S.E. = 0.049
			Time = 88
				Rate = 0.94
				S.E. = 0.042
				Time = 205

	100	Rate = 0.92
		S.E. = 0.033
		Time = 78
			Rate = 0.94
			S.E. = 0.03
			Time = 148
				Rate = 0.945
				S.E. = 0.029
				Time = 323

	200	Rate = 0.94
		S.E. = 0.031
		Time = 150
			Rate = 0.945
			S.E. = 0.029
			Time = 298
				Rate = 0.962
				S.E. = 0.025
				Time = 548

Scenario #2: Two Circles

Monte Carlo Simulation		*# of Species, S*	
		5	**15**

# of Sites, n	**50**	Rate = 0.68 	
		S.E. = 0.118
		Time = 49
			Rate = 0.81
			S.E. = 0.071
			Time = 150

	150	Rate = 0.672
		S.E. = 0.142
		Time = 149
			Rate = 0.98
			S.E. = 0.015
			Time = 285

Scenario #3: Cubic

Monte Carlo Simulation		*# of Species, S*	
		5	**15**

# of Sites, n	**50**	Rate = 0.88 	
		S.E. = 0.071
		Time = 51
			Rate = 0.60
			S.E. = 0.106
			Time = 147

	150	Rate = 0.85
		S.E. = 0.115
		Time = 220
			Rate = 0.973
			S.E. = 0.032
			Time = 431

Scenario #4: Linear/Quadratic/Cubic

Monte Carlo Simulation		*# of Species, S*	
		8	**24**

# of Sites, n	**75**	Rate = 0.78 	
		S.E. = 0.074
		Time = 244
			Rate = 0.853
			S.E. = 0.043
			Time = 687

	225	Rate = 0.907
		S.E. = 0.035
		Time = 842
			Rate = 0.922
			S.E. = 0.05
			Time = 2422

Unexpected results were found for Scenario #2 (Two Circles) using 150 sites and 5 species, where the median success rate was lower than that of the trials with 50 sites and 5 species. Likewise, for Scenario #3 (Cubic) using 50 sites and 15 species, we find the unexpected result that using 50 sites and 15 species results in a much smaller median success rate than using 50 sites and 5 species. These findings are not fully understood, but could be a result of the small number of Monte Carlo iterations.

Table 4.3.3: MC results comparing various combinations of *n* and *S* that satisfy $n \times S = 2400$.

n	600	300	200	150	120	100	80	60	50	40	30
S	4	8	12	16	20	24	30	40	48	60	80
MC Median Success Rate	0.949	0.952	0.96	0.94	0.95	0.94	0.95	0.95	0.96	0.95	0.933
MC Standard Error	0.024	0.021	0.027	0.021	0.031	0.025	0.026	0.025	0.028	0.038	0.045
MC Median Computation Time (s)	338	291	298	340	366	402	474	629	721	1014	1455

4.3.3 The Choice of *n* and *S*

In Section 3.2.3 I highlighted the fact that an increase in *S* increases the amount of available data at the same time that it increases the number of parameters that must be estimated. In this section, I address the effect of this tradeoff upon the classification success rate and computation time.

To accomplish this, I further investigate the Baseline Linear scenario with all its standard simulation assumptions about true parameter generation and starting parameter value choices. I fix $V = n \times S$, the total number of species presence/absence observations to be 2400 and examine the results of choosing different values for *n* and *S* that satisfy this constraint. Again, a Monte Carlo simulation is used to average out the variation due to the random sample of sites. Table 4.3.3 shows the results, in terms of median classification success rate, standard error, and computation time, for eleven different *n*, *S* combinations.

We find remarkably consistent results among the different combinations of number of sites and number of species except when the number of species is particularly large relative to the number of sites. In that case, performance is worse as measured by the standard error. This makes intuitive sense because with this few sample locations it becomes difficult to draw a sample that represents all the features of the true habitat maps. The same difficulty does not appear to be true for involving increasingly fewer species, at least through *S*=4. Despite this increase in standard error for trials with fewer sites, success rates are generally quite high. The fact that even a scenario with just 30 sites can deliver a high success rate, provided there are enough species to compensate, offers flexibility that is encouraging for the context of real world research.

Perhaps it is not surprising that there is some counterbalancing between the number of sites and number of species present in the scenario. With more sites, we gain more data directly informing us about the location of the habitats. On the other hand, more species observations directly helps in the task of defining and distinguishing these habitats in terms of their characterizing species. These effects seem roughly equal.

4.3.4 Model Selection and Misspecification

In this section I present an exercise in model selection based on covariate polynomial order. Until this point, the form of the logit-linear model for π has been chosen to be the same form as the model from which the simulation data were generated except for the brief investigation in Section 4.2.5. Here I apply a standard model selection technique to evaluate model misspecifation.

My approach is to estimate the Linear/Quadratic/Cubic scenario (simulating 24 species at each of 225 sites across the study region) by fitting linear, quadratic, cubic, quartic, and quintic models for the effects in the additive logistic exponent. The data sample used is the sample corresponding to the Monte Carlo median classification success rate in Section 4.2.4. For each of the five model forms, I measure the classification success rate and the log likelihood value corresponding to the estimated parameters in ***A*** and ***B***. Using the log likelihood values, I also calculate associated the Akaike information criterion (AIC) and the small sample corrected AIC version (AICc) defined as (Akaike, 1973; Hurvich & Tsai, 1989):

$$AIC = 2k - 2\ln(L) \qquad and \qquad AICc = AIC + \frac{2k(k+1)}{N-k-1}$$

where $k = p(H-1) + S(H+1)$ is the number of parameters and $N = n \times S$ is the total number of observed data points. Lower values of these statistics indicate better models.

Table 4.3.4: Covariate model selection results for Scenario #4 (Linear/Quadratic/Cubic).

Model Order	Log Likelihood Value	AIC	AICc	Success Rate
Linear	-3050	6358	6365	0.680
Quadratic	-2974	6218	6225	0.871
Cubic	-2957	6195	6203	0.924
Quartic	-3099	6491	6500	0.876
Quintic	-2960	6225	6234	0.924

Results of the model selection experiment based on covariate polynomial order for this scenario are presented in Table 4.3.4. As we expected, the cubic model has the lowest AIC and AICc, the highest log likelihood, and is tied for the highest success rate among all model orders considered. The fact that we have identified the correct model order suggests that standard model selection techniques are appropriate and useful when fitting this model.

Figure 4.3.1 shows the classification plots resulting from each of the covariate polynomial order models listed in Table 4.3.4. In this set of maps we find visual evidence to corroborate our previous model selection decision based on log likelihood and AIC. Each of the MLE maps in this figure show reasonably good habitat estimation within the limitations of the polynomial order used (for example, the linear version can produce only a single linear boundary in the individual logistic habitat probability plot). The logit-linear model, while badly erring in terms of habitat shape and completely omitting Habitat 1 (black), at least locates Habitats 2 (blue), 3 (red) and 4 (green) reasonably. The quadratic version locates all habitats well, but still lacks the sufficient degrees of freedom to estimate the shapes of Habitats 1 and 2 correctly. The cubic MLE is clearly the most accurate fit to the true scenario, successfully estimating the difficult Habitat 2 and providing reasonably good fits for Habitats 1 and 3. The 4th and 5th order models show that in this case overfitting the model allows for more accurate map estimation than underfitting it, but certainly does not perform as well as fitting the correct cubic model. It is also worth noting that the red habitat (defined by the quadratic parameters in the underlying model) is best estimated by the quadratic model, and the blue cubic habitat (defined by the cubic parameters) is best estimated by the cubic model.

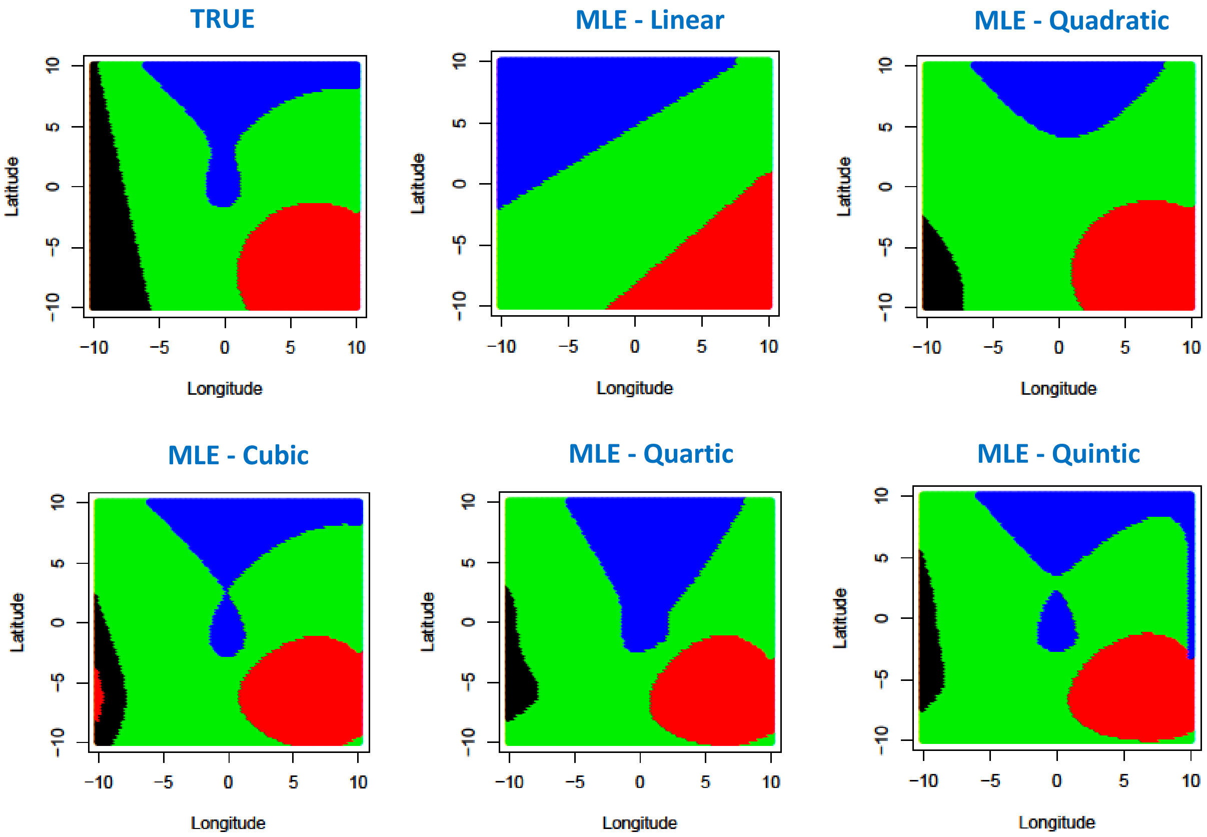

Figure 4.3.1: Habitat classification plots for models postulating various covariate polynomial orders for Scenario #4 (Linear/Quadratic/Cubic).

4.3.5 Misspecification of *H* and the Potential for Empirical Selection of *H*

In a real world habitat estimation situation, a researcher will not necessarily know or wish to assume the true number of habitats represented within a study region. This uncertainty motivates the need for an approach to compare models that include different numbers of habitats for a given scenario. In other words, we can consider how to choose *H*. In theory, model selection techniques similar to the previous section may aid in addressing how many habitats should be included. However, a researcher might not base his/her choice of *H* entirely on the results of this type of model selection since he/she will incorporate their expert knowledge about the subject.

Table 4.3.5: Model selection metrics for several estimation attempts of the Four Circles scenario using different numbers of modeled habitats.

# Modeled Habitats	Log Likelihood Value	AIC	AICc
3	-7550	15360	15363
4	-7555	15440	15445
5	-6957	14315	14322
6	-6918	14306	14316
7	-6795	14131	14143

In the following exercise I model the five-habitat Scenario #6 (Four Circles) with models postulating 3, 4, 5, 6 and 7 habitats. The standard setting of 30 species at each of 400 sampled sites (see Section 4.2.6) is used. Also, I used the same specific random sample of sites that was used before. Because the true number of habitats and estimated number of habitats differ in four of these estimation attempts, we lose the ability to calculate and compare classification success rates. For example, if we model 7 habitats to a 5 habitat system, there is no direct way to draw a one-to-one correspondence between the 5 true and 7 estimated habitats. Thus we must rely on comparing log likelihood values and visually assessing estimated habitat maps.

The table above presents log likelihood values and Akaike information criteria for the habitat estimation attempts which model each of three through seven habitats. In this case, all three criteria lead us to choose the model that involves seven habitats, which contrasts with the five habitats that we know to truly exist in this study region. Development of an appropriate model selection criterion or method to better balance explanatory power and parsimony in selecting the number of habitats in a system will be left as an avenue for future research.

In the context of this experiment, we can still compare the results to the truth to evaluate the potential impact of misspecification. To do this, we can examine Figure 4.3.2 which presents the true classification plot alongside classification plots corresponding to models that estimate 3, 4, 5, 6 and 7 habitats. However, a consequence of our inability to calculate the classification success rate is that we also lose our ability to swap habitats into their ‘correct’ color labels. Thus when viewing the maps in Figure 4.3.2, we should consider the shapes of the habitats but not the colors.

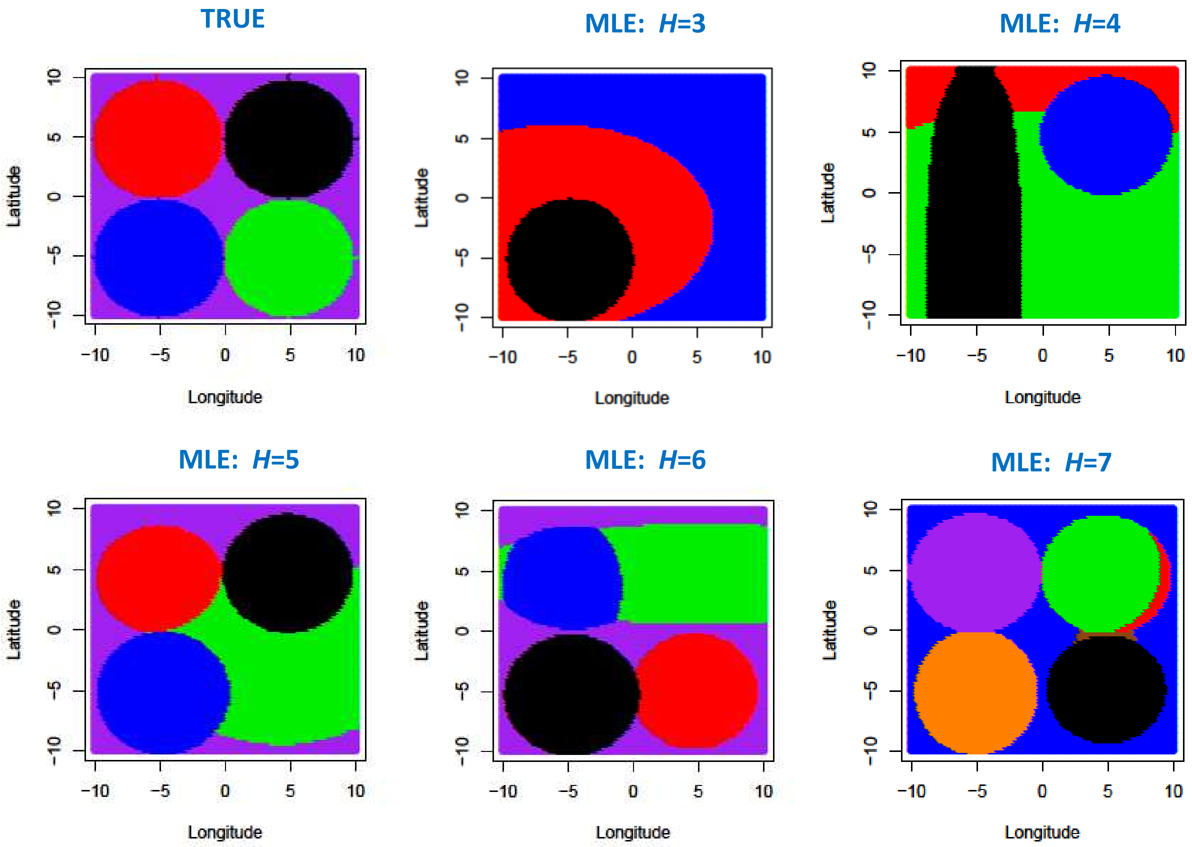

Figure 4.3.2: True habitat classification plot and estimated habitat classification plots for models postulating 3, 4, 5, 6, and 7 habitats for Scenario #6 (Four Circles).

Before discussing Figure 4.3.2, we must note a subtle difference between this experiment and the straightforward estimation of the Four Circles scenario presented in Section 4.2.6: the difference in size between the true and estimated *B* matrix necessitates a different strategy for choosing parameter starting values. For simplicity, I choose to set all parameter starting values to zero. This starting value choice is not a favorable one, resulting in relatively poor estimation. From better starting values, Section 4.2.6 shows better performance.

Understanding that habitat colors, like habitat labels, are arbitrary, we compare only the habitat shapes, sizes and locations of the five MLE plots to the true classification plot. The model with *H*=3 successfully estimates shape and location of the bottom left habitat, while the other three circular habitats are not accurately fit. In the classification plot for the model with *H*=4, the upper right habitat is estimated successfully and the two leftmost circular habitats are lumped together in to one MLE habitat. The

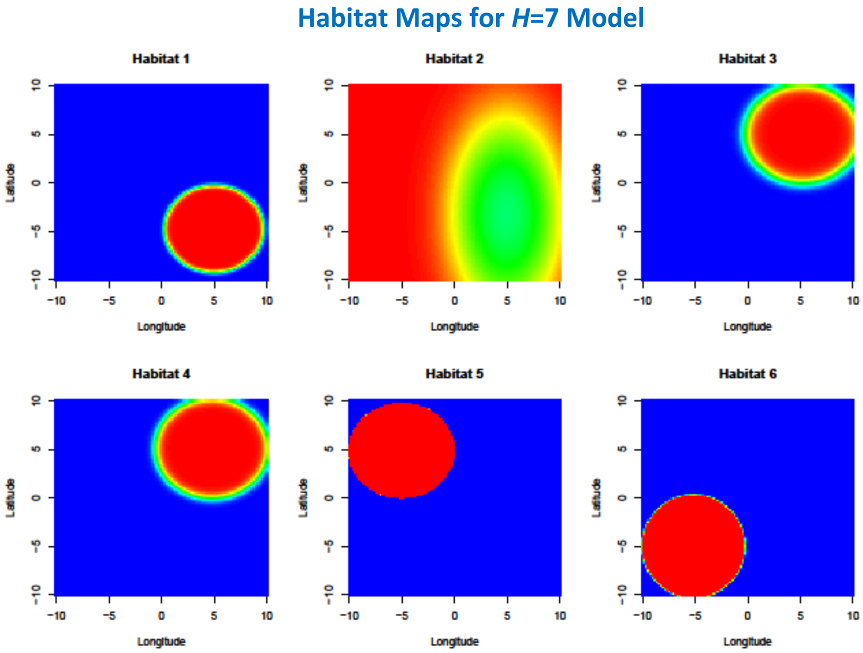

Figure 4.3.3: Estimated individual logistic habitat probability plots from model postulating *H*=7 for 5-habitat Scenario #6 (Four Circles).

model with *H*=5 provides for a much more successful habitat classification plot. Here we see that the two leftmost circular habitats and the upper right circular habitat are estimated closely, while the bottom right habitat is too large. For the model with *H*=6, one habitat is dominated by the other five habitats at every location within the study region, thus only five of the habitats claim any region of the classification plot.

The final classification plot representing the model with *H*=7 is interesting in that we see evidence that the two circular habitats in the rightmost quadrants are fringed by tiny, superfluous habitats (red and brown). To investigate this further, consider Figure 4.3.3 which shows the estimated individual logistic habitat probability plots. We see that the probability maps for Habitats 3 and 4 are nearly identical. Meanwhile, because of the almost universal coverage of the first six habitats, the indirectly estimated Habitat 7 will be able to claim the tiny brown sliver represented in the final plot of Figure 4.3.2. My tentative conclusion is that it is better to fit large *H,* then ignore any small or trivial habitats that result.

After inspecting the habitat classification plots for various modeled *H* values, we can better assess the model selection exercise conducted above. Neither standard model selection techniques (log likelihood, AIC, AICc), nor comparing true to estimated habitat classification plots advocate matching the modeled

H with the true H. It seems we want the modeled value for H to be at least as big as the true value of H. In a sense, AIC provides the wrong answer by suggesting that we model H too high. However, this answer is in agreement with a qualitative assessment of the estimated classification plots.

In the context of real world application, scientists often have a sense of first the reasonable range of the number of habitats, and second, limitations on the reasonable and scientifically meaningful sizes of possible habitats, or both. It may even be possible to synthesize several of the habitat maps into one understanding of the underlying true habitat structure. For instance, by combining the information found in the classification plots for only the 5 and 6 habitat models, we have strong evidence that there are two circular habitats occupying the left-hand side of the study region, and evidence—albeit less strong—that two circular habitats reside on the right-hand side as well. Further consideration for the selection of the quantity of modeled habitats is left to individual users of the model, and to future study.

4.3.6 Analysis of Species Prevalences

In Section 3.2.4 I hypothesized that habitats might be easier to distinguish when rare species are very rare and common species very common. This section will further examine that hypothesis.

To do so, I again experimented with the Baseline Linear habitat scenario. As elsewhere in this thesis, the sample of 20 species and 200 sites that generated the Monte Carlo median success rate is used. I ran six trials using this linear scenario and these data, each time changing only the magnitude of the τ_{jh} values which (with α_j) define the abundance of species j in habitat h. The method described in 3.2.4 for randomly attributing positive and negative τ values to species within habitats during the data simulation step applies here. The smallest τ values I chose are ±log(1.1). This generates a common species presence probability of 0.52 and a rare species presence probability of 0.48. Thus, this case provides very little information about habitat differentiation. The largest τ values I chose are ±log(10). This generates a presence probability contrast of 0.91/0.09 for the common and rare species respectively, thereby providing extremely strong discriminatory power.

Table 4.3.3 displays the classification success rates for each of the six trials along with τ values, the corresponding odds for the presence of common species, and the associated common/rare species presence probability contrasts. Inspecting the column of classification success rates, we find that this

Table 4.3.3: Comparison of classification success rates for six different values of τ for Scenario #1 (Baseline Linear).

Trial #	Tau	Odds for Common Species	Common/Rare Species Presence Probabilities	Classification Success Rates
1	±log(1.1)	11:10	0.52/0.48	0.855
2	±log(1.5)	3:2	0.60/0.40	0.855
3	±log(2)	2:1	0.67/0.33	0.91
4	±log(3)	3:1	0.75/0.25	0.965
5	±log(5)	5:1	0.83/0.17	0.945
6	±log(10)	10:1	0.91/0.09	0.975

metric generally increases with the magnitude of τ. This evidence seems to support my initial intuition that habitats are easier to distinguish when the probability disparity between common and rare species within a set of habitats is large. It must be noted that these success rates are subject to some variation based on site sample. I have not controlled for this variation with Monte Carlo simulation because of computing limitations.

4.3.7 Algorithm Convergence

This section will address the topic of convergence of the optimization algorithm used to find the maximum likelihood estimates of the model parameters. In Section 2.3 I presented the relative convergence criterion (RCC) used to halt the estimation algorithm when the following condition is satisfied:

$$RCC_B = \max_{k,h}[abs(Q_{kh})] < \delta \qquad and \qquad RCC_A = \max_{p,h}\left[abs\left(R_{ph}\right)\right] < \delta$$

Where Q_{kh} is the k, h element of $\frac{B_{new}-B_{previous}}{B_{previous}+\varepsilon}$ and R_{ph} is the p, h element of $\frac{A_{new}-A_{previous}}{A_{previous}+\varepsilon}$.

I set $\varepsilon=10^{-5}$ which ensures against dividing by zero, and δ is set to 10^{-6} for the examples in this paper. Not surprisingly, this threshold is found to be more difficult to achieve for models with larger numbers of parameters than for simpler problems. Because this algorithm was found to converge fairly rapidly and clearly, monitoring of the RCC was not crucial, though retaining this stopping criterion did help to conserve computation time especially during Monte Carlo simulation.

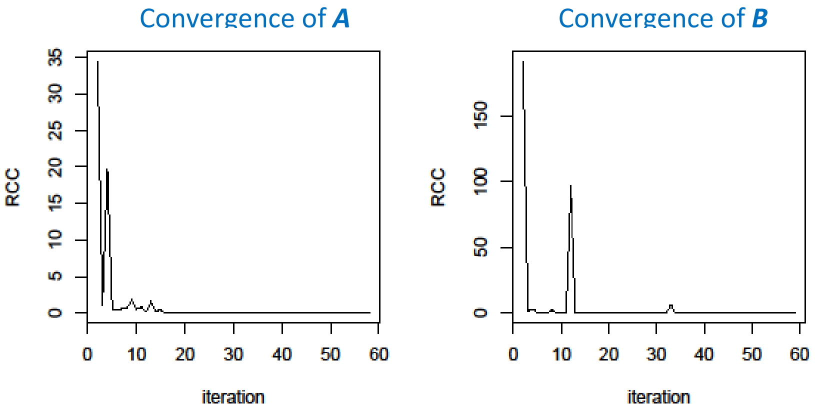

Figure 4.3.4: Relative Convergence Criterions across iterations for estimated ***A*** and ***B*** matrices.

A more tangible understanding of the algorithm's convergence can be obtained by viewing the plots in Figure 4.3.4. Here, the convergence criterion is plotted across the 61 algorithm iterations required for Scenario #1 (Baseline Linear) to reach the convergence condition shown above. The jump made in the first iteration is by far the largest, and is omitted to prevent stretching the vertical axis too much. Most change in both the ***A*** and ***B*** parameter matrices can be seen to occur in the first few iterations of the algorithm. Beyond that, we observe several small spikes and one relatively large change in the ***B*** matrix around the twelfth iteration.

Evidence that optimization can converge rapidly can also be seen by examining the evolution of the individual logistic habitat probability plots and classification plots generated by the set of parameter matrices updated at the end of each iteration. These plots are presented in Figure 4.3.4 for the estimation of Scenario #3 (Cubic). I set all parameter starting values to zero, thus resulting in featureless initial habitat maps. We can see features in the probability surface begin to emerge after the first iteration, and the maps already look reasonable by the second or third iteration. The estimated habitat maps achieve roughly their final form by the fifth or sixth iteration.

Note that in order to compare these habitat estimates with the true maps, we would need go through the additive logistic model, and swap habitats for proper colors. Rather than doing all this, we can inspect the evolution of the habitat classification plots, which automatically take this process into account. The result is shown in Figure 4.3.5. The classification plots seem to converge to their final appearance—and to a good estimate of the true classification plot—by about the fifth or sixth iteration in this example.

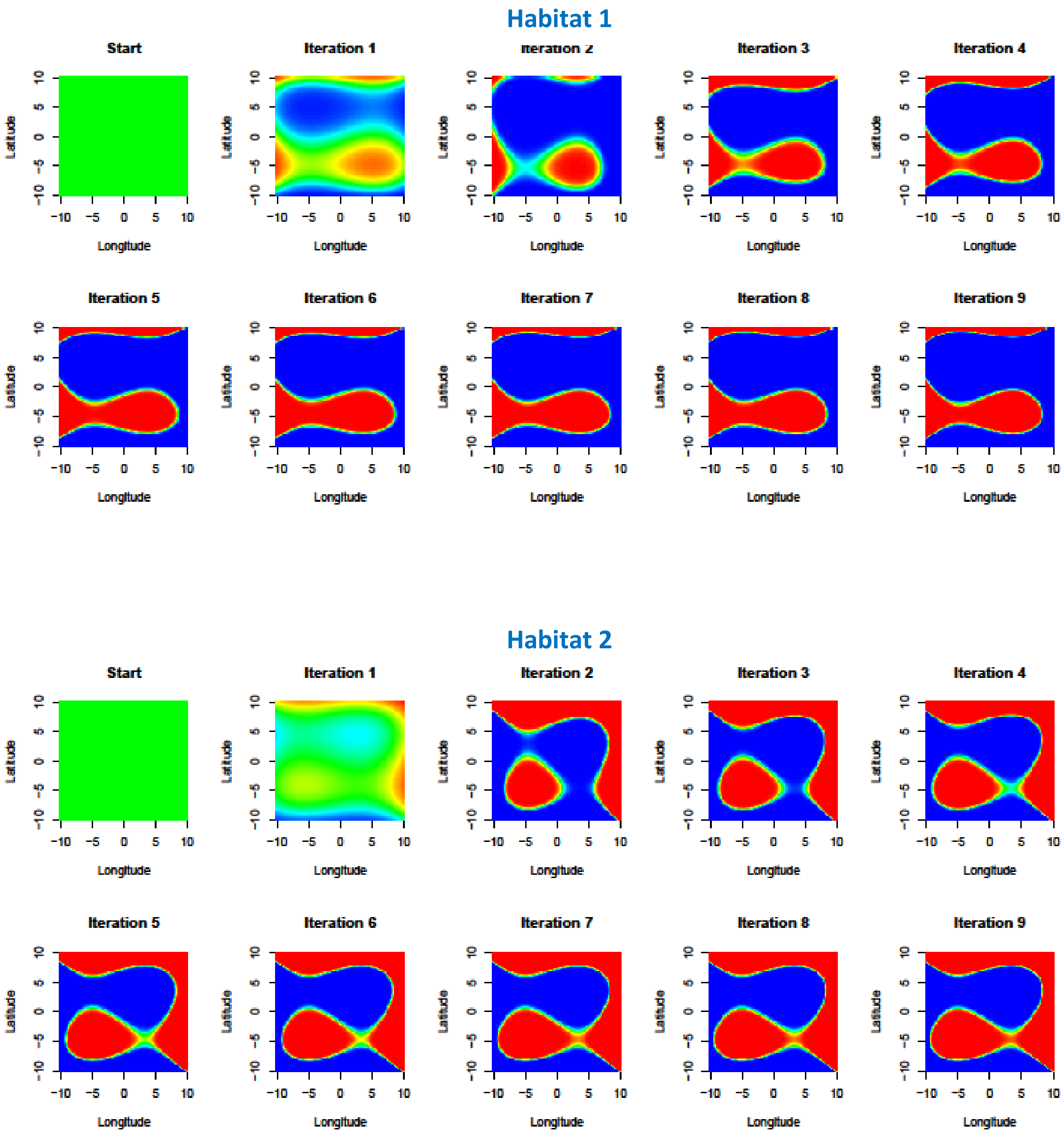

Figure 4.3.4: Evolution of the individual logistic habitat probability maps for Habitat 1 (top two rows) and Habitat 2 (bottom two rows) through nine iterations of MLE optimization. In each case, iterations progress from left to right and then top to bottom.

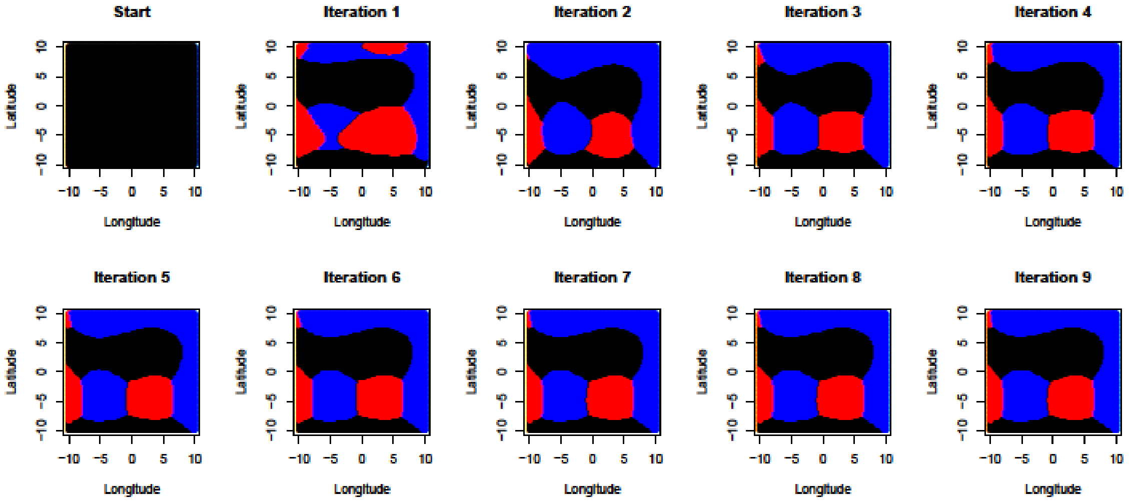

Figure 4.3.5: Evolution of the habitat classification plot through 9 iterations of MLE optimization. Iterations progress from left to right and then top to bottom.

4.3.8 Analysis of Starting Value Strategies for Optimization

Because of the complexity of the likelihood function, a challenge in likelihood maximization is finding the true global optimum rather than a local one. In many cases, the choice of starting values at which the nonlinear blocked Gauss-Seidel algorithm begins is important. In this section, I explore several strategies for generating parameter starting values and the effect that this choice has on our ability to uncover the (presumably) global maximum of the likelihood function.

I consider three general strategies for generating parameter starting values. These strategies are implemented and compared with respect to the estimation of the familiar Linear Baseline scenario. As elsewhere in this thesis I use the sample of 20 species at 200 sites that produced the Monte Carlo median success rate. The three methods for generating starting values are described below. First, it is useful at this point to recall the true parameter values we attempt to estimate in this scenario.

$$\boldsymbol{B}^T = \begin{bmatrix} 0 & 0.4 & -0.2 \\ 0 & 0 & 1 \end{bmatrix}_{(H-1)\times p} \qquad \textit{and} \qquad \boldsymbol{A} = \begin{bmatrix} 0 & \pm\log(3) & \pm\log(3) & \sum_h \tau_{1h} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \pm\log(3) & \pm\log(3) & \sum_h \tau_{Sh} \end{bmatrix}_{S\times H}$$

Method #1: Random (Uniform) Starting Values

For the first method, I draw random values uniformly across a specified range for $A_{0,jh}$ and $B_{0,kh}$, the values that populate the ***A*** and ***B*** starting parameter value matrices, respectively. Several different ranges from which to draw are compared in the results table in this section.

Method #2: Fixed Starting Values

The second strategy simply sets

$$A_{0,jh} = B_{0,kh} = c \quad \forall\, j,k,h$$

where *c* is a constant scalar. This method, like the previous one, incorporates no prior knowledge of the true parameter values themselves, nor of their values relative to one another.

Method #3: 'True Values + Noise' Starting Values

The final method generates parameter starting values that are informed by the true parameters that were originally used to generate the true scenario. Thus this method requires an educated guess of the true parameter values. This strategy would be particularly applicable if, for instance, we had access to the fitted model pertaining to the same study region from a previous analysis and wanted to update the estimated habitat maps with new data. However, from a practical point of view, the purpose here is merely to hasten convergence in some of my estimation attempts. For this method, I generate starting values by randomly drawing from the following distributions.

$$A_{0,jh} \sim Normal\left(A_{jh}\,, abs(\frac{A_{jh}}{SNR})\right)$$
$$B_{0,kh} \sim Normal\left(B_{kh}\,, abs(\frac{B_{kh}}{SNR})\right)$$

where SNR is a signal to noise ratio. The SNR can be chosen to allow for more simulated uncertainty in the choice of starting parameter values. Using this framework, a small SNR will generally allow for high variability around the true parameter values, while a large SNR should produce starting parameter values that are near to the true parameter values. I compare values of SNR that range from 0.1 to 10 below.

Table 4.3.4 contains success rates and log likelihood values for experiments utilizing several variations on each of the three defined methods. The most obvious result in this table is that most trials result in a log likelihood value near -2388 (highlighted in green), and that among those trials, success rates are high and only vary between 0.945 and 0.97. A second important observation is that each case that uses informed starting values with high SNRs—which should provide an advantage in accurate estimation of parameters—agrees upon the same -2389 log likelihood value. These two notes provide compelling evidence that -2388 does in fact correspond to the global maximum of the likelihood function.

Three pathological results exist among these trials and are highlighted in other colors. The trial for which I randomly chose starting values uniformly over the broadest range, [-10,10], and the trial for which the most noise was added to the starting values (SNR=0.1) find unsatisfactory local modes with lower log likelihood values and success rates. It is important to note that these results are from an experiment that was run only once. It is likely that better results could be achieved in these three cases by overlaying a 'random starts' approach, i.e. repeating the optimization many times for (random) diverse starting values and choosing the best result. This very simple approach is based on hoping that the global optimum is found in at least one case. The multiple starts help increase the chances of getting at least one success.

Table 4.3.4: Classification success rate and log likelihood value measurements for 18 trials broken into 3 different methods for generating starting parameter values.

Method	**Submethod**	**Success Rate**	**Log Likelihood Value**
Uninformed, Random (Uniform) with range:	[-0.5,0.5]	0.945	-2390
	[-1,1]	0.955	-2390
	[-2,2]	0.96	-2389
	[-5,5]	0.96	-2389
	[-10,10]	0.755	-2455
Uninformed, fixed starting value:	-1	0.97	-2389
	-0.5	0.96	-2389
	0	0.825	-2473
	0.5	0.96	-2389
	1	0.97	-2389
Informed, with SNR:	0.1	0.745	-2497
	0.5	0.96	-2389
	1	0.97	-2389
	2	0.965	-2389
	3	0.97	-2388
	4	0.97	-2388
	5	0.97	-2389
	10	0.965	-2388

Thus I have shown that the estimation procedure can succeed with disparate sets of starting parameter values, finding the unique, globally optimal estimate in most cases.

4.3.9 Analysis of Habitat Boundary Width

Throughout the examples in this paper, habitat estimation has consistently generated overly sharp habitat boundaries. In Section 3.3 I introduced the idea that habitat boundaries can be made sharper or gentler simply by multiplying $\boldsymbol{B}$ by a scalar. Through Section 4.2 we observed $\widehat{\boldsymbol{B}}$ matrices that appeared to be, essentially, scalar multiples of $\boldsymbol{B}$. In this section, I investigate the hypothesis in Section 4.2.4, namely that habitats with gentler boundaries tend to be estimated with gentler boundaries.

To address this, I did 250 Monte Carlo replicated estimates for each of the three parameter matrices shown below. The only difference between the three matrices is the scalar factor controlling the sharpness of the probabilistic habitat boundaries. I then tested whether the $\widehat{\boldsymbol{B}}$'s retained evidence that could distinguish which true $\boldsymbol{B}$ they originated from.

$$\boldsymbol{B}_{normal}^{\mathrm{T}} = \begin{bmatrix} -12.75 & 5.10 & -0.51 & 5.10 & -0.51 \\ -12.75 & -5.10 & -0.51 & -5.10 & -0.51 \end{bmatrix}$$

$$\boldsymbol{B}_{sharp}^{\mathrm{T}} = 8 \times \begin{bmatrix} -12.75 & 5.10 & -0.51 & 5.10 & -0.51 \\ -12.75 & -5.10 & -0.51 & -5.10 & -0.51 \end{bmatrix}$$

$$\boldsymbol{B}_{gentle}^{\mathrm{T}} = \frac{1}{8} \times \begin{bmatrix} -12.75 & 5.10 & -0.51 & 5.10 & -0.51 \\ -12.75 & -5.10 & -0.51 & -5.10 & -0.51 \end{bmatrix}$$

To quantify the amount by which $\widehat{\boldsymbol{B}}_d$—the *d*th replicated estimate of one of the above $\boldsymbol{B}$ matrices—misestimated the true habitat sharpness, I calculated a set of individual parameter scalars, $c_{d,kh}$. These values are explicitly defined by the following relation:

$$B_{kh} = c_{d,kh}\widehat{B}_{d,kh}, \qquad k\text{=1…}p,\ h\text{=1…(}H\text{-1) and d=1…250}$$

where the *k* and *h* subscripts index the rows and columns, respectively, of the $\boldsymbol{B}$ matrices. Thus, for each of the 250 Monte Carlo replications I obtained a matrix of scalar (i.e. multiplicative) errors between $\boldsymbol{B}$ and $\widehat{\boldsymbol{B}}$, $\boldsymbol{c}_{d,p\times H}$. I then found the median of these scalars within each of these matrices, and denoted it as C_d, d=1…250. Finally, I let C denote the median of the 250 C_d's. The C values for the three boundary sharpness scenarios are presented in Table 4.3.5 along with the median classification success rate.

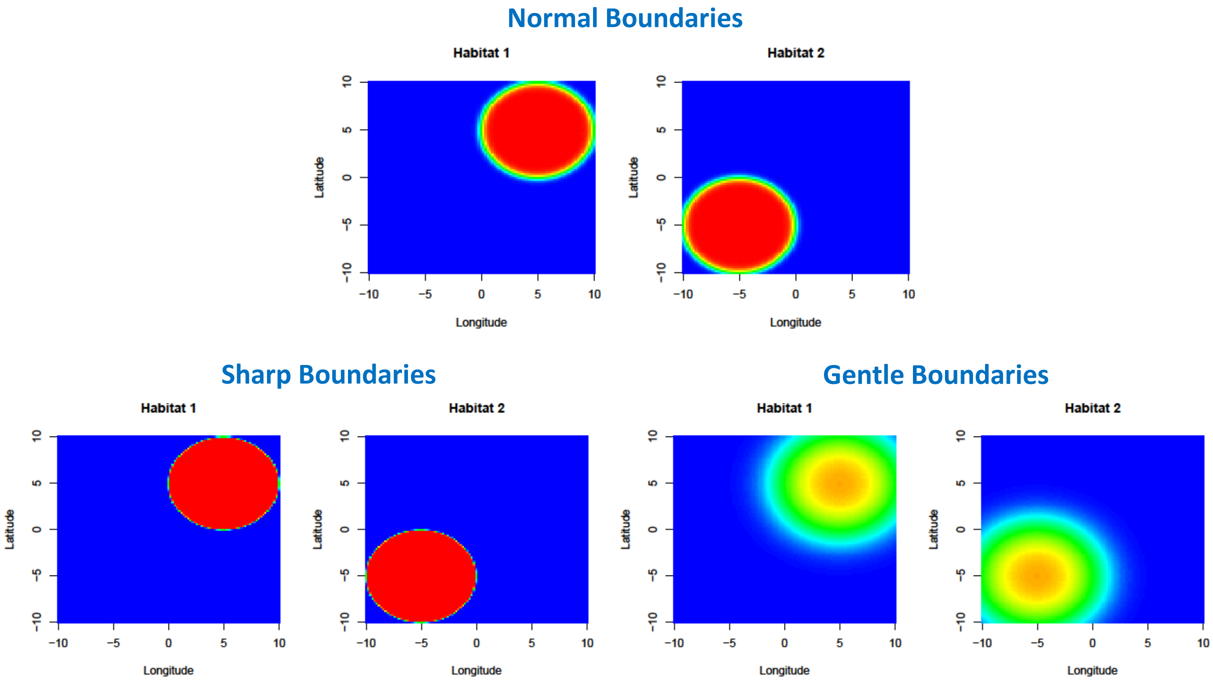

Figure 4.3.6: True individual logistic habitat probability plots displaying normal, sharp and gentle boundaries for Scenario #2 (Two Circles).

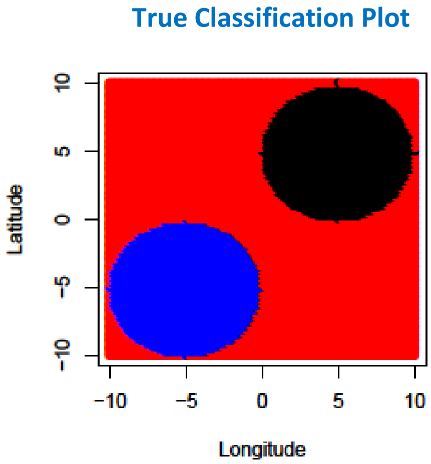

Figure 4.3.7: True habitat classification plot common to all three scenarios presented in Figure 4.3.6.

The true individual logistic habitat probability plots corresponding to the three parameter matrices are presented in Figure 4.3.6. The visible difference in habitat boundary width is a result of the scalar factor difference of 8 between the gentle and normal scenario and between the normal and sharp scenario. Despite this difference, all three of these ***B*** matrices produce the same true habitat classification plot, seen in Figure 4.3.7.

Table 4.3.5: Median scalar errors between $\boldsymbol{B}$ and $\widehat{\boldsymbol{B}}$ and median classification success rates for 250 Monte Carlo replications for each of three boundary sharpness variations of Scenario #2 (Two Circles).

Boundary Width	Overall Median $\boldsymbol{B}$ Scalar, C	Median Classification Success Rate
Gentle	463.251	0.873
Normal	98.045	0.980
Sharp	5.097	0.993

Table 4.3.5 shows an important difference in classification success rates between the habitats with gentle boundaries and those with normal or sharp boundaries. The gently sloping habitat probability surfaces are more difficult to estimate than steeper surfaces, resulting in a classification success rate of only 0.873. This is likely because there is closer competition between Habitat 3 and the other two habitats in the gentle boundary sharpness variation.

Now we consider the scalar factors in Table 4.3.5. The C value under the normal scenario is nearly 20 times the C value under the sharp boundary scenario. Knowing that $\boldsymbol{B}_{normal}^{\mathrm{T}}$ and $\boldsymbol{B}_{sharp}^{\mathrm{T}}$ differ by a scalar factor of only 8, we can say that on average, estimation of the normal boundary scenario actually attains sharper boundaries than the estimates of the sharp boundary scenario, since 20 is larger than 8. The reason for these results is not well understood, but it provides evidence that is contradictory to our hypothesis that habitats with gentler boundaries tend to be estimated with gentler boundaries. Rather, it seems that this model is unable to detect the true steepness of habitat probability surfaces effectively at all; excessively sharp boundaries result in each case. This topic is left to further study.

It is worth noting that despite this failure to distinguish the boundary sharpnesses of the normal and sharp scenario variations, the median classification success rates are extremely high for both. This fact is direct evidence that the habitat classification plot is estimated exceptionally accurately and consistently among the 250 MC replicated estimations.

CHAPTER 5: CONCLUSIONS & FUTURE WORK

5.1 Summary

This paper has investigated the performance of a statistical model developed in Foster et al. (2011) to estimate habitat maps from species presence-absence information and environmental covariate data. The model is characterized simultaneously by a parameter matrix ***B*** parameters which determines the shape and location of habitats in a study region, and by a parameter matrix ***A*** which determines the underlying species presence probabilities that are assumed to define habitats and distinguish them from one another.

A collection of simulation testing scenarios were invented and species and covariate data was simulated from each. Scenarios ranged from simple to complex, including: between three and nine habitats, between fifteen and forty characterizing species, between 150 and 400 sites, and polynomial coefficients in the covariate effects up to cubic terms. Estimation proved to be very successful for each of the habitat scenarios presented. This conclusion was evidenced by diagnostic plots of the estimated parameters, comparisons between true and estimated habitat probability plots and habitat classification plots, and by considering the habitat classification success rates.

Each of these scenarios was estimated using the maximum likelihood method and a blocked non-linear Gauss-Seidel optimization algorithm. The algorithm was found to clearly and quickly converge within five or six iterations for all of the estimation exercises included in the paper. However, the computation time required by the estimation algorithm was often demanding, with the nine habitat scenario requiring six hours to complete nine iterations.

Many additional investigations addressed several special topics related to the performance of the model and estimation algorithm. Monte Carlo simulation was used to compare the relationship between estimation performance, number of sites samples, and number of species while while controlling and studying the variation due to the random site locations. When considering Monte Carlo replications, this computational cost is sometimes prohibitively expensive.

An exercise in covariate model selection showed that comparison of AIC and AICc values is a viable method for selecting the polynomial order of the covariate effects. Misspecification of the number of habitats was also studied. This was another form of model selection. Visual inspection of estimated habitat probability plots and qualitative scientific reasoning were argued to be important components of a strategy to choose an appropriate number of habitats rather than relying exclusively on AIC values.

An analysis comparing species prevalences was conducted confirming that when the difference in presence probabilities between rare and common species grows, habitats are more distinguishable from one another and habitat estimation is more successful.

As with any complex optimization problem, the choice of parameter starting values can be important. It was shown that for this model, the optimization algorithm would sometimes converge to a local maximum of the likelihood function rather than the global optimum when using poor starting values. This was rarely a problem, however, when reasonable parameter starting values were chosen. Users of this model should consider a random starts local search approach to find a good set of parameter starting values, especially for more complex scenarios where the parameter matrix ***B*** is very large.

Lastly, an investigation was conducted into effective estimation of habitat boundary widths. It was shown that the model has a lot of trouble accurately estimating boundary sharpness, even though boundary locations are estimated well. Throughout this paper, the examples illustrate this tendency. The reason for this issue is not presently known and is left to be investigated in future study.

Despite the boundary sharpness misestimation issue, the model described in Foster, et al. (2011) combined with the estimation algorithm described in this paper have the capability to estimate complex habitat scenarios very accurately. Estimation results for the largest, most complex scenario presented in the paper (Scenario #7 is specified by 400 parameters) are impressive.

5.2 Future Work

Having successfully estimated this assortment of simple and complicated habitat scenarios, future work with this model and estimation algorithm should introduce more complexity, and ultimately real data. A good first step toward increasing model complexity (while still working with simulated data that can be objectively tested) is to introduce new covariates. While all examples in this paper simply used spatial coordinates as the covariate variables or as proxies for covariate variables (see Section 3.1), introducing covariates with their own underlying spatial structure (e.g. high shipping traffic areas, ocean currents, high pollutant areas, etc.) will allow for more naturally shaped habitats.

The Great Barrier Reef data set mentioned in Section 2.2 includes data for 13 covariates. This real life scenario is much more complex than any of the testing scenarios presented in this paper, especially after allowing for polynomial and interaction terms among these variables. Therefore, future work with this model should use a higher quantity of covariates to evaluate how estimation results change with greatly increased covariate complexity. Additionally, this would allow for much a more in depth practice in model selection.

Scenario #7 (Diamond/Circles), whose estimation results are presented in Section 4.2.7, includes n=400 sites, S=40 species and H=9 habitats. Meanwhile, the Great Barrier Reef data set includes n=1200 sites, S=200 species and 15-20 habitats. While increasing n and S to these levels will increase computation requirements, examples in this paper have shown that an increase in these values will improve model estimation. The more challenging step, which will require future investigation, is to increase the number of habitats to 15 or 20. Moreover, eight of the habitats in Scenario #7 are simplified in the sense that they are each connected sets. Future work should investigate more complex scenarios where a single habitat type can be found in many disjoint locations across the study region.

Perhaps the largest unresolved question left by this paper is how to estimate habitat boundary transitions properly. Sections 4.2 and 4.3.9 demonstrate the tendency for overly precise boundary estimates. Clearly, future work is needed to address this problem. One possible method to investigate this would be to sample more sites along the (expected) habitat boundary regions. Although this is not a remedy to the problem, it might help formulate some hypotheses about how to adjust the model. It may be useful to introduce a 'tuning parameter' to the model that is a scalar multiplier of the $\boldsymbol{B}$ parameter matrix. The exact placement of this parameter in the model framework is unclear, since it must be both effective and identifiable/estimable.

Finally, future work with this model should explore potential biological/ecological interpretations of the typical results. For example, the idea of a habitat mosaic (defined as an area comprised of multiple habitat types) may be applicable where multiple habitats have high individually computed probabilities of existence in a single region. It would also be worthwhile to connect this framework to the idea of prime and fringe habitats. Finally, it would be interesting to consider 3-dimensional habitats existing, for instance, at different ocean depths (surface, reef, deep sea) but at the same latitude/longitude coordinate.

5.3 Reflections on my Master's Project

Although my project involved a variety of efforts—both theoretical and applied—the lion's share of my research time was spent programming in R to investigate questions about the model. Through this practice, I learned a tremendous amount of the R programming language. Specifically, I learned how to build code that is as general as possible (to cope with the ever-changing components of simulation testing), and the importance of this style of programming. For example, my code is written to handle a general number of sites, species, habitats, and a general covariate polynomial model form. This sounds trivial, but, as described below, this was a very challenging issue because it involved writing code that would automatically write additional code tailored specifically to the habitat scenario at hand.

One of the most challenging specific tasks required for this research was to enable my code to generate an arbitrary number of conditional likelihood functions—one for each of the separately estimated blocks in a single iteration of the blocked non-linear Gauss-Seidel estimation algorithm[4]. This was necessary because the *optim()* R function used within the algorithm for the estimation of each parameter block requires a uniquely named likelihood function which needed to be written slightly differently for each block. A general outline of the code that wrote S unique likelihood functions corresponding to the S blocks in the $\boldsymbol{A}$ parameter matrix is shown here.

```
for (s in 1:S) {
    assign ( paste ( log likelihood function name ),
        function ( $\boldsymbol{A}$ block, $\boldsymbol{A_u}$, $\boldsymbol{B_u}$, covariate data, presence/absence data ) {;
            $\boldsymbol{A_u}$[s,] = $\boldsymbol{A}$ block
            rest of log likelihood function
        }
    )
}
```

[4] Recall that there are S+H-1 blocks for a given scenario.

where ***A*** *block* is the chunk optimized for log-likelihood, all else constant, and $\boldsymbol{A_u}$ and $\boldsymbol{B_u}$ are the most recently updated sets of parameter estimates for the ***A*** and ***B*** parameter matrices respectively. Note that I use red text to denote the items whose names change depending on block. The likelihood functions are then called through *optim()* within the estimation algorithm with code like:

optimal *A* block = optim(*A* block, eval(as.name(paste(log likelihood function name))),…)

By using the same combination of *assign()*, *paste()*, *eval()* and *as.name()*, a similar set of code was used to generate a likelihood function tailored to optimize each of the blocks within the ***B*** matrix.

Generality for the modeled number of habitats and modeled covariate polynomial order was also incorporated to allow for misspecification of *H* and *p*. My code was also built to easily compare various values for τ_j parameters, various random samples of sites, and various strategies for generating parameter starting values among other testing questions. At every point along the way, it was important to produce informative diagnostic plots and measurements to evaluate the simulation-estimation process, and to organize the vast amount of estimation results effectively. Also, I learned that user-friendliness is a very important aspect of good code writing. Developing my code in this regard was helpful for myself as I ran hundreds of simulations, and will be important as I pass on the code to those who will use it next. The appendix provides annotated code and an overview of the functionality of the code.

In the process of researching and writing this paper, I learned a tremendous amount about investigating statistical models, statistical computing, and organizing and communicating complex results and interpretations. This was also my first experience working with a mixture model, here used to allow the probabilistic membership of sites to habitats. Also, the additive logistic function used in the model to transform a linear combination of covariate data and coefficient parameters to the probability scale was new to me. The cornerstone of my contribution to the understanding of this model was the development of sophisticated testing scenarios, the coding of data generation and, particularly, model estimation procedures and the evaluation of estimation performance. Consequently, I now know this process well.

REFERENCES

Habitat Mosaic. (2009, February 9). Retrieved October 11, 2011, from Biology Online: http://www.biology-online.org/dictionary/Habitat_mosaic

Aitchison, J. (1982). The Statistical Analysis of Compositional Data. *The Journal of the Royal Statistical Society - Series B*, 139-177.

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. *Proceedings of the Second International Symposium of Information Theory.* Budapest: Akedemia Kiaodo.

Foster, S. D., Givens, G. H., Dunstan, P. K., & Darnell, R. (2011). *A Model for Estimating Habitat Maps from Species Presence-Absence and Environmental Covariate Data.*

Givens, G. H., & Hoeting, J. A. (2005). *Computational Statistics.* Hoboken: Wiley.

Hurvich, C. M., & Tsai, C.-L. (1989). Regression and time serios model selection in small samples. *Biometrika 76*, 297-307.

Kedem, B., & Fokianos, K. (2002). *Regression Models for Time Series Analysis.* Hoboken: Wiley Series in Probability and Statistics.

McLachlan, G., & Peel, D. (2000). *Finite Mixture Models.* John Wiley and Sons Inc.

APPENDIX

Overview of Code

In the following appendices I have included the principal R code used to run habitat simulation-estimation exercises. This code is broken into four broad sections, each written to address a specific task. They are designed to be run sequentially and take a habitat estimation example from initialization all the way to post-estimation evaluation and habitat maps. The four sections of code are introduced here.

1. **Testing Scenario Setup**
   - Initializes sample size, algorithm options, plotting and save options, etc.
   - Sets up scenario – ***B*** and ***A*** matrices.
2. **Function Library**
   - Writes different 'flavors' of the likelihood function recursively depending on *S* and *H*.
   - Defines function to simulate presence/absence data.
   - Several miscellaneous functions.
3. **Parameter Estimation**
   - Samples sites and simulates presence/absence observations.
   - Plots and saves true habitat maps.
   - Generates optimization starting values and initializes updatable parameter arrays.
   - Runs blocked nonlinear Gauss-Seidel likelihood optimization algorithm to estimate model parameters.
   - Performs habitat swapping to correctly label estimated habitats.
   - Computes model selection criteria.
   - Outputs estimation results to save folder as .csv files.
4. **Plots and Diagnostics**
   - Creates plots to directly compare parameters to their estimates.
   - Creates habitat probability plots (individual and logistic).
   - Creates habitat classification plots (for sampled sites and across grid)

Appendix A – Testing Scenario Setup

##

This collection of code sets up a simulation testing scenario in terms of underlying habitats (B matrix, A matrix), data/sample (n, S), optimization preferences (max iterations, convergence threshold, optimization method, etc.), plotting/saving options, and model misspecification. The items in the first segment are individually described. The second segment defines the B matrix and calculates the H and p values which are implicit within this matrix. The final segment allows for misspecification of covariate polynomial order and number of habitats, and is optional. This code must be run first when doing a simulation-estimation exercise. The setup for Scenario #1 is presented with descriptions of each input item, and the setups for Scenarios #2-#7 are included subsequently.

```
######################################################################################
                                 ##  SCENARIO #1  ##

NAME="BASELINE LINEAR EXAMPLE"          # scenario name, for save pathname
n=200                                   # number of sampled sites
S=20                                    # number of species
m=9                                     # maximum algorithm iterations
inits=2                                 # SNR for starting parameter value creation
seed=26                                 # random number seed for site sample
tauseed=2011                            # random number seed for simulated tau values
tauspread=log(3)                        # tau magnitude
method="Nelder-Mead"                    # optimization method within optim()
delta=0.000001                          # convergence threshold
pixels=101                              # resolution of maps
rawpoly=T                               # raw or orthogonalized covariate polynomials
trueplots=T                             # should true maps be drawn?
saveplots=F                             # should maps/excel files be saved?
savepath="pathname"                     # for creation of save folder

######################################################################################

B1=c(0,2,-1)*(20/pixels)                # habitat 1 B parameters
B2=-c(0,0,-5)*(20/pixels)               # habitat 2 B parameters
B.true=rbind(B1,B2)
p.true=p.model=ncol(B.true)             # assume correct covariate specification
H.true=H.model=nrow(B.true)+1           # assume correct H specification
parscale=c(1,100,300)                   # parameter scale for optim()

######################################################################################

# Model Misspecification
# p.model=3 ; parscale=rep(1,p.model)   # allows for  p.true ≠ p.model
# H.model=3                             # uses 0-starting values if H.true ≠ H.model

######################################################################################
```

```
######################################################################################
######################################################################################
                                  ##  SCENARIO #2  ##

NAME="TWO CIRCLES EXAMPLE"
n=150
S=15
m=9
inits=2
seed=13
tauseed=2011 ; tauspread=log(3)
method="Nelder-Mead"
delta=0.000001
pixels=101
rawpoly=T
trueplots=T ; saveplots=T
savepath="pathname/"

 B1=c(-1.27513012,0.51025615,-0.05102561,0.51025615,-0.05102561)*10
 B2=c(-1.27513012,-0.51025615,-0.05102561,-0.51025615,-0.05102561)*10
 B.true=rbind(B1,B2)
 p.true=p.model=ncol(B.true) ; H.true=H.model=nrow(B.true)+1
 parscale=rep(1,p.model) ; parscale=c(25,10,1,10,1)

# Model Selection
# p.model=5 ; parscale=rep(1,p.model)
# H.model=3

######################################################################################
######################################################################################
                                  ##  SCENARIO #3  ##

NAME="CUBIC EXAMPLE"
n=150
S=15
m=9
inits=2
seed=2
tauseed=2011 ; tauspread=log(3)
method="Nelder-Mead"
delta=0.000001
pixels=101
rawpoly=T
trueplots=T ; saveplots=T
savepath="pathname/"

 B1=-c(-10,-10,.40,.20,-40,.07,.7)
 B2=c(-15,-10,.40,.20,-2,.3,.08)
 B.true=rbind(B1,B2)
 p.true=p.model=ncol(B.true) ; H.true=H.model=nrow(B.true)+1
 parscale=rep(1,p.model) ; parscale=c(100,100,4,2,150,1,1)

# Model Selection
# p.model=7 ; parscale=rep(1,p.model)
# H.model=3

###############################################################################################
###############################################################################################
                                      ##  SCENARIO #4  ##

NAME="LIN-QUAD-CUBE EXAMPLE"
n=225
S=24                                                                       # 30
m=9                                                   ################### 25
inits=2                                      ###   ??????   ### 0
seed=35                                      ################### 2011
tauseed=2011 ; tauspread=log(3)                          # log(3)
method="Nelder-Mead"                                              # NM
delta=0.000001                                                    # 0.000001
pixels=101                                                                 # 101
rawpoly=T
trueplots=T ; saveplots=T
savepath="pathname/"

 B1=-c(75,10,0,0,2,0,0)*(20/pixels)
 B2=-c(-75,5,40,-3,-10,20,-4)*(6/pixels)
 B3=c(-1.78028927-0.5,0.50875790,-0.03633985,0,-0.50875790,-0.03633985,0)*15
 B.true=rbind(B1,B2,B3)
 p.true=p.model=ncol(B.true) ; H.true=H.model=nrow(B.true)+1
 parscale=rep(1,p.model) ; parscale=c(20,10,2,1,10,2,1)

# Model Selection
# p.model=7 ; parscale=rep(1,p.model)
# H.model=4

###############################################################################################
###############################################################################################
                                      ##  SCENARIO #5a  ##
NAME="DIAMOND (Linear) EXAMPLE"
n=200
S=20
m=9
inits=2
seed=2
tauseed=2011 ; tauspread=log(3)
method="Nelder-Mead"
delta=0.000001
pixels=101
rawpoly=T
trueplots=T ; saveplots=T
savepath="pathname/"

 B1=-c(9,1,-1)*(120/pixels)
 B2=c(-9,1,-1)*(120/pixels)
 B3=-c(9,-1,-1)*(120/pixels)
 B4=c(-9,-1,-1)*(120/pixels)
 B.true=rbind(B1,B2,B3,B4)
 p.true=p.model=ncol(B.true) ; H.true=H.model=nrow(B.true)+1
 parscale=rep(1,p.model) ; parscale=c(10,1,1)

# Model Selection
# p.model=3 ; parscale=rep(1,p.model)
# H.model=5

#####################################################################################
#####################################################################################
                                        ##  SCENARIO #5b  ##

NAME="DIAMOND (Quad) EXAMPLE"
n=200
S=20
m=9
inits=2
seed=2
tauseed=2011 ; tauspread=log(3)
method="Nelder-Mead"
delta=0.000001
pixels=101
rawpoly=T
trueplots=T ; saveplots=T
savepath="pathname/"

 B1=-c(9,1,0,-1,0)*(120/pixels)
 B2=c(-9,1,0,-1,0)*(120/pixels)
 B3=-c(9,-1,0,-1,0)*(120/pixels)
 B4=c(-9,-1,0,-1,0)*(120/pixels)
 B.true=rbind(B1,B2,B3,B4)
 p.true=p.model=ncol(B.true) ; H.true=H.model=nrow(B.true)+1
 parscale=rep(1,p.model) ; parscale=c(10,1,1,1,1)

# Model Selection
# p.model=5 ; parscale=rep(1,p.model)
# H.model=5

#####################################################################################
#####################################################################################
                                        ##  SCENARIO #6  ##

NAME="FOUR CIRCLES EXAMPLE"
n=400
S=30
m=9
inits=2
seed=5
tauseed=2011 ; tauspread=log(3)
method="Nelder-Mead"
delta=0.000001
pixels=101
rawpoly=T
```

```
trueplots=T ; saveplots=T
savepath="pathname/"

 B1=c(-1.27513012,0.51025615,-0.05102561,0.51025615,-0.05102561)*10
 B2=c(-1.27513012,-0.51025615,-0.05102561,-0.51025615,-0.05102561)*10
 B3=c(-1.27513012,-0.51025615,-0.05102561,0.51025615,-0.05102561)*10
 B4=c(-1.27513012,0.51025615,-0.05102561,-0.51025615,-0.05102561)*10
 B.true=rbind(B1,B2,B3,B4)
 p.true=p.model=ncol(B.true) ; H.true=H.model=nrow(B.true)+1
 parscale=rep(1,p.model) ; parscale=c(25,10,1,10,1)

# Model Selection
# p.model=5 ; parscale=rep(1,p.model)
# H.model=5


################################################################################
################################################################################
                                   ##  SCENARIO #7  ##

NAME="DIAMOND-CIRCLES EXAMPLE"
n=400
S=40
m=9
inits=2
seed=1
tauseed=2011 ; tauspread=log(3)
method="Nelder-Mead"
delta=0.000001
pixels=101
rawpoly=T
trueplots=T ; saveplots=T
savepath="pathname/"

 B1=-c(9,1,0,-1,0)*(120/pixels)
 B2=c(-9,1,0,-1,0)*(120/pixels)
 B3=-c(9,-1,0,-1,0)*(120/pixels)
 B4=c(-9,-1,0,-1,0)*(120/pixels)
 B5=c(-2.8428571,2.5714286,-0.4285714,2.5714286,-0.4285714)
 B6=c(-1.4428571,-2.5714286,-0.4285714,-2.5714286,-0.4285714)
 B7=c(-2.1428571,-2.5714286,-0.4285714,2.5714286,-0.4285714)
 B8=c(-2.1428571,2.5714286,-0.4285714,-2.5714286,-0.4285714)
 B.true=rbind(B1,B2,B3,B4,B5,B6,B7,B8)
 p.true=p.model=ncol(B.true) ; H.true=H.model=nrow(B.true)+1
 parscale=rep(1,p.model) ; parscale=c(10,5,1,5,1)

# Model Selection
# p.model=5 ; parscale=rep(1,p.model)
# H.model=9
```

Appendix B – Function Library

```
################################################################################
```

This collection of code defines the functions that will later be used for data simulation and estimation. S+H-1 versions of the likelihood function are required by the blocked non-linear Gauss-Seidel algorithm, and thus must be defined conditional on knowing S and H. Consequently, this section must be run after the ‘Testing Scenario Setup’ section of code. The various ‘flavors’ of coded likelihood function are specified in the first two segments of code below.

The artificial.data2() function simulates presence absence data across the n sites for S species. These are obtained by Bernoulli draws with success probabilities according to the mixture model of species presence probabilities defined by alpha and tau values weighted by habitat membership probabilities. The function outputs a binary matrix of presence/absence ‘observations’ where 1 represents presence and 0 represents absence.

Lastly, a few miscellaneous functions and one required R package are included which are used at various points of the code that follows.

```
################################################################################
#                    LOG-LIKELIHOOD FUNCTIONS FOR EACH 'BLOCK' (for optim)      #
################################################################################

## NOTE: at.mat is the A parameter matrix described in the paper including alphas and taus.
## optim() requires that the block of parameters (at.piece, B.piece) to optimize for each
## small optimization be the first argument of the function.  Thus a unique function must
## be written for each of the S+H-1 blocks, each allowing the corresponding parameter block
## to to be input separately and then gluing that block into the larger parameter matrix.

for (s in 1:S) {
  assign( paste("loglike.at",s,sep='') , function(at.piece,at.mat,B.mat,Xdata,Ydata) { ;
        at.mat[s,1:H.model]=at.piece ;
        at.mat[s,H.model+1]=-sum(at.mat[s,2:H.model]) ;
        Xdata=as.matrix(Xdata) ;
        Ydata=as.matrix(Ydata) ;
        n=nrow(Xdata) ;
        S=nrow(at.mat) ;
        bin=array(NA,c(n,S,H.model)) ;
        for (j in 1:S) { ;
                logitmu=at.mat[j,1]+at.mat[j,2:(H.model+1)] ;
                mu=exp(logitmu)/(1+exp(logitmu)) ;
                for (i in 1:H.model) { bin[,j,i]=ifelse(as.logical(Ydata[,j]),mu[i],1-mu[i]) } ;
        } ;
        b=matrix(NA,nrow=n,ncol=H.model) ;
        for (k in 1:H.model) { b[,k]=apply(bin[,,k],1,prod.fun) } ;
        expXB=exp(pmin(pmax(Xdata%*%t(B.mat),-100),100)) ;
        M=matrix(NA,nrow=n,ncol=H.model) ;
        M[,1:(H.model-1)]=expXB/(1+apply(expXB,1,sum)) ;
        M[,H.model]=1-apply(M[,1:(H.model-1)],1,sum) ;
        sitelike=apply(b*M,1,sum) ;
```

```
            llval=sum(log(sitelike)) ;
            list(llval=llval) ;
            }
  )
}
################################################################################
################################################################################
for (h in 1:H.model) {
  assign( paste("loglike.B",h,sep='') , function(B.piece,at.mat,B.mat,Xdata,Ydata) { ;
            B.mat[h,]=B.piece ;
            Xdata=as.matrix(Xdata) ;
            Ydata=as.matrix(Ydata) ;
            n=nrow(Xdata) ;
            S=nrow(at.mat) ;
            bin=array(NA,c(n,S,H.model)) ;
            for (j in 1:S) { ;
                        logitmu=at.mat[j,1]+at.mat[j,2:(H.model+1)] ;
                        mu=exp(logitmu)/(1+exp(logitmu)) ;
                        for (i in 1:H.model) { bin[,j,i]=ifelse(as.logical(Ydata[,j]),mu[i],1-mu[i]) } ;
            } ;
            b=matrix(NA,nrow=n,ncol=H.model) ;
            for (k in 1:H.model) { b[,k]=apply(bin[,,k],1,prod.fun) } ;
            expXB=exp(pmin(pmax(Xdata%*%t(B.mat),-100),100)) ;
            M=matrix(NA,nrow=n,ncol=H.model) ;
            M[,1:(H.model-1)]=expXB/(1+apply(expXB,1,sum)) ;
            M[,H.model]=1-apply(M[,1:(H.model-1)],1,sum) ;
            sitelike=apply(b*M,1,sum) ;
            llval=sum(log(sitelike)) ;
            list(llval=llval) ;
            }
  )
}
################################################################################


################################################################################
#                                   ARTIFICIAL.DATA2() FUNCTION                                   #
################################################################################
artificial.data2 <- function (H.form,parms,dat,S,alpha,tau) {
            ## length(alpha)=S, dim(tau)= S,H+1
            link.fun <- make.link("logit")
            H <- nrow(parms)+1 # number of habitats 1 more that specified in glm form
            mu <- tau + alpha
            for(i in 1:(S*(H)))
                        mu[i] <- link.fun$linkinv(mu[i]) ## logit link on mu
            X <- model.matrix(H.form,dat)
            p.habitat <- matrix(NA,dim(X)[1],H)
            for(h in 1:(H-1))
                        p.habitat[,h] <- (pmin(pmax(X%*%parms[h,],-100),100))
            ## glm form for H-1 habitats -> additive.logistic give H habitats
            for(i in 1:dim(X)[1])
                        p.habitat[i,] <- additive.logistic(p.habitat[i,-H])
            p <- matrix(NA,dim(X)[1],S)
            set.seed(1234)
            for(i in 1:dim(X)[1])
                        p[i,] <- mu%*%p.habitat[i,] ## mu H rows long, p.habitat H cols
            sample <- rbinom(dim(p)[1]*dim(p)[2],1,p)
            dim(sample) <- c(dim(X)[1],S)
            colnames(sample) <- colnames(p) <- paste("S.",1:S,sep="")
            list(sample=sample,p=p,phab=p.habitat,X=X,mu=mu) # output sample from binomial & probabilities of
species
}
########################################################################################



########################################################################################
#                                    ADDITIVE.LOGISTIC() FUNCTION                                    #
########################################################################################
additive.logistic <- function (x,inv=FALSE)
{
 if(inv){
   x <- log(x/x[length(x)])
   return(x)
  }
  x.t <- exp(x)
  x.t <- x.t/(1+sum(x.t))
  x.t[length(x.t)+1] <- 1-sum(x.t)
  return(x.t)
}
########################################################################################


########################################################################################
#                                      MISCELLANEOUS FUNCTIONS                                      #
########################################################################################

prod.fun <- function(x) {
                        exp(sum(log(x)))
}

invlogit=function(x) { exp(x)/(1+exp(x)) }

########################################################################################
```

Appendix C – Parameter Estimation

###

This collection of code samples sites, simulates data and estimates model parameters using a blocked nonlinear Gauss-Seidel algorithm. If the user chooses to plot the true habitat maps and save estimation results (saveplot and trueplots options), this section of code does these tasks.

Before estimation, parameter starting values are chosen based on the user's preferences specified in the setup of the scenario. After the estimation procedure, habitat swapping is done to give the best possible set of labels to the habitats, the classification success rate is computed, and the log likelihood is used to compute AIC and AICc values. Finally, relevant results are output as .csv files into the save folder.

This set of code must be run after the Testing Scenario Setup and Function Library collections of code.

```
#################################################################################

## create folder to save plots and .csv results
if (saveplots) {
  time=Sys.time()
  newpath=paste(savepath,NAME,"  ",format(time, "%a %b %d %Y  %H.%M.%S %Z"),sep="")
  dir.create(newpath)
}

#####################################

## create folder to save plots and .csv results
if (saveplots) {
  time=Sys.time()
  newpath=paste(savepath,NAME,"  ",format(time, "%a %b %d %Y  %H.%M.%S %Z"),sep="")
  dir.create(newpath)
}

#####################################

## max polynomial effect in covariates, used to create data sample
ord.model=(p.model-1)/2
ord.true=(p.true-1)/2

#####################################

## colors for habitat plotting
colors=c("black","blue","red","green2","purple",
         "darkorange1","chocolate4","gray47","gray75",
         "blue4","purple4","red4","deeppink1","slateblue3")

#####################################

## Sample sites, compute corresponding covariate data (lat, long, and poly's)
set.seed(seed)
x=runif(n,-10,10) ; z=runif(n,-10,10)
```

```
if (exists("rawpoly") && rawpoly) {
        dat.sample <- data.frame(y=1,poly(x,ord.true,raw=T),poly(z,ord.true,raw=T))
        form.data <- y~poly(x,ord.true,raw=T)+poly(z,ord.true,raw=T)                    # Raw Polynomials
} else {dat.sample <- data.frame(y=1,poly(x,ord.true,raw=F),poly(z,ord.true,raw=F))
        form.data <- y~poly(x,ord.true,raw=F)+poly(z,ord.true,raw=F) }                  # Orthogonal
Polynomials

## Generate 'true' alpha and tau values
set.seed(tauseed)
alpha.true=rep(0,S)
taubits=c(tauspread,-tauspread)
tau.true=matrix(sample(taubits,size=S*(H.true-1),replace=T),nrow=S)
tau.h=-apply(tau.true,1,sum)
tau.true=cbind(tau.true,tau.h)
at.true=cbind(alpha.true,tau.true)

## Simulate species presence/absence with Bernoulli trials
test <- artificial.data2(form.data,B.true,dat.sample,S,alpha.true,tau.true)


if (trueplots) {
#############################################################################
#                              CREATE HABITAT PLOTS                          #
#############################################################################

## Create data (lat and long) at each point in a grid across the study region
xx=seq(-10,10,length.out=pixels) ; zz=seq(-10,10,length.out=pixels)
dat.grid <- data.frame(xx,zz)
dat.grid <- expand.grid(dat.grid$xx,dat.grid$zz)
dat.grid <- data.frame(y=1,dat.grid)                    ## create grid for map
names(dat.grid) <- c("y","xx","zz")
if (exists("rawpoly") && rawpoly) {
        form.data2 <- y~poly(xx,ord.true,raw=T)+poly(zz,ord.true,raw=T)
} else { form.data2 <- y~poly(xx,ord.true,raw=F)+poly(zz,ord.true,raw=F) }
X.true <- model.matrix(form.data2,dat.grid)

#############################################################################
               ##  Individual Logistic Habitat Probability Plots  ##

## 'Individual' logistic probabilities (see paper) for each point in grid
link.fun <- make.link("logit")
myprobs.indiv.true=matrix(NA,nrow=pixels^2,ncol=H.true-1)
for (h in 1:(H.true-1)) {
        myprobs.indiv.true[,h] <- link.fun$linkinv(X.true%*%B.true[h,])
}
indivprobs.grid.true <- array(NA,c(pixels,pixels,H.true-1))
for (i in 1:length(xx)) {
  for (j in 1:length(zz)) {
    for(k in 1:(H.true-1)) {
        indivprobs.grid.true[i,j,k] <- myprobs.indiv.true[,k][which(dat.grid$xx==xx[i] & dat.grid$zz==zz[j])]
    }
  }
}
```

```
## Define plot layout
if (H.true==2) { par(mfrow=c(1,2),oma=c(0,0,2,0))
} else if (H.true==3)  { windows(width=9,height=5) ; par(mfrow=c(1,2),oma=c(0,0,2,0))
} else if (H.true==4)  { windows(width=12,height=4.5) ; par(mfrow=c(1,3),oma=c(0,0,2,0))
} else if (H.true==5)  { windows(width=13,height=4) ; par(mfrow=c(1,4),oma=c(0,0,2,0))
} else if (H.true==6)  { windows(width=12,height=9) ; par(oma=c(0,0,2,0)) ;
layout(matrix(c(1,1,2,2,3,3,1,1,2,2,3,3,0,4,4,5,5,0,0,4,4,5,5,0), 4, 6, byrow=TRUE))
} else if (H.true==7)  { windows(width=12,height=9) ; par(mfrow=c(2,3),oma=c(0,0,2,0))
} else if (H.true==8)  { windows(width=13,height=7.5) ; par(oma=c(0,0,2,0)) ;
layout(matrix(c(1,1,2,2,3,3,4,4,1,1,2,2,3,3,4,4,0,5,5,6,6,7,7,0,0,5,5,6,6,7,7,0), 4, 8, byrow=TRUE))
} else if (H.true==9)  { windows(width=13,height=7.5) ; par(mfrow=c(2,4),oma=c(0,0,2,0))
} else if (H.true==10) { windows(width=12,height=15) ; par(mfrow=c(3,3),oma=c(0,0,2,0))
} else if (H.true==11) { windows(width=15,height=6.5) ; par(mfrow=c(2,5),oma=c(0,0,2,0))
} else if (H.true==12) { windows(width=13,height=10) ; par(oma=c(0,0,2,0)) ;
layout(matrix(c(1,1,2,2,3,3,4,4,1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,5,5,6,6,7,7,8,8,0,9,9,10,10,11,11,0,0,9,9,10,10,11,11,0),
6, 8, byrow=TRUE))
} else if (H.true==13) { windows(width=13,height=10) ; par(mfrow=c(3,4),oma=c(0,0,2,0))
} else { print("Matrix Layout not predifined for this number of Habitats") }

## plot individual probs for each habitat (& indicate where samples are taken)
for (l in 1:(H.true-1)) {
	image(xx,zz,indivprobs.grid.true[,,l],xlab="Longitude",ylab="Latitude",main=paste("Habitat",l),zlim=c(0,1),c
ol=rev(rainbow(100,end=4/6)))
#	points(x,z,col="white",cex=4,pch='.')
}
title("True Individual Habitats",outer=T,line=-0.5,cex.main=2,font.main=4,col.main="blue")

if (saveplots) { dev.copy2pdf(file=paste(newpath,"/_True Indiviual Habitat Maps",".pdf",sep="")) }


################################################################################
			##  Additive Logistic Habitat Probability Plots  ##

## 'Additive' logistic probabilities (see paper) for each point in grid
myprobs.joint.true=t(apply(pmin(pmax(X.true%*%t(B.true),-100),100),1,additive.logistic))
jointprob.grid.true <- array(NA,c(pixels,pixels,H.true))
for (i in 1:length(xx)) {
  for (j in 1:length(zz)) {
    for(k in 1:(H.true)) {
	jointprob.grid.true[i,j,k] <- myprobs.joint.true[,k][which(dat.grid$xx==xx[i] & dat.grid$zz==zz[j])]
    }
  }
}

## Define plot layout
if (H.true==2) { par(mfrow=c(1,2),oma=c(0,0,2,0))
} else if (H.true==3)  { windows(width=12,height=4.5) ; par(mfrow=c(1,3),oma=c(0,0,2,0))
} else if (H.true==4)  { windows(width=13,height=4) ; par(mfrow=c(1,4),oma=c(0,0,2,0))
} else if (H.true==5)  { windows(width=12,height=9) ; par(oma=c(0,0,2,0)) ;
layout(matrix(c(1,1,2,2,3,3,1,1,2,2,3,3,0,4,4,5,5,0,0,4,4,5,5,0), 4, 6, byrow=TRUE))
} else if (H.true==6)  { windows(width=12,height=9) ; par(mfrow=c(2,3),oma=c(0,0,2,0))
} else if (H.true==7)  { windows(width=13,height=7.5) ; par(oma=c(0,0,2,0)) ;
layout(matrix(c(1,1,2,2,3,3,4,4,1,1,2,2,3,3,4,4,0,5,5,6,6,7,7,0,0,5,5,6,6,7,7,0), 4, 8, byrow=TRUE))
} else if (H.true==8)  { windows(width=13,height=7.5) ; par(mfrow=c(2,4),oma=c(0,0,2,0))
} else if (H.true==9)  { windows(width=12,height=15) ; par(mfrow=c(3,3),oma=c(0,0,2,0))
} else if (H.true==10) { windows(width=15,height=7) ; par(mfrow=c(2,5),oma=c(0,0,2,0))
} else if (H.true==11) { windows(width=13,height=10) ; par(oma=c(0,0,2,0)) ;
layout(matrix(c(1,1,2,2,3,3,4,4,1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,5,5,6,6,7,7,8,8,0,9,9,10,10,11,11,0,0,9,9,10,10,11,11,0),
6, 8, byrow=TRUE))
} else if (H.true==12) { windows(width=13,height=10) ; par(mfrow=c(3,4),oma=c(0,0,2,0))
} else { print("Matrix Layout not predifined for this number of Habitats") }

## plot each habitat (& indicate where samples are taken)
for (l in 1:(H.true)) {
		image(xx,zz,jointprob.grid.true[,,l],xlab="Longitude",ylab="Latitude",main=paste("Habitat",l),zlim=c(0,1),co
l=rev(rainbow(100,end=4/6)))
#		points(x,z,col="white",cex=4,pch='.')
}
title("True Additive Habitats",outer=T,line=-0.5,cex.main=2,font.main=4,col.main="blue")

if (saveplots) { dev.copy2pdf(file=paste(newpath,"/_True Additive Habitat Maps",".pdf",sep="")) }

#####################################################################################
					##  Habitat Classification Plot  ##

windows(width=6,height=7)
par(mfrow=c(1,1),oma=c(0,0,2,0))
votfun=function(x) {
  ismax=x==max(x)
  ((1:H.true)[ismax])[1]  #uniqueness ensured
  }
winner.true=apply(myprobs.joint.true,1,votfun)
plot(xx,zz,type="n",xlab="Longitude",ylab="Latitude",main="True")
points(dat.grid[,2],dat.grid[,3],col=colors[winner.true],pch=16)
#points(x,z,col="white",cex=4,pch='.')
title("Classification Plot",outer=T,line=-0.5,cex.main=2,font.main=4,col.main="blue")

if (saveplots) { dev.copy2pdf(file=paste(newpath,"/_True Classification Plot",".pdf",sep="")) }

#####################################################################################
}

#####################################################################################
#					LIKELIHOOD OPTIMIZATION STEP					#
#####################################################################################

epsilon=.00001
w.conv=m

## Create Xdata at each of the previously sampled sites according to the
## MODELED polynomial order in the covariates (this may differ from true order)
dat.sample.trueorder <- as.matrix(data.frame(y=1,poly(x,ord.true,raw=T),poly(z,ord.true,raw=T)))
if (p.model != p.true) {
if (exists("rawpoly") && rawpoly) {
```

```
            dat.sample <- data.frame(y=1,poly(x,ord.model,raw=T),poly(z,ord.model,raw=T))
            form.data <- y~poly(x,ord.model,raw=T)+poly(z,ord.model,raw=T)                              # Raw Polynomials
} else {dat.sample <- data.frame(y=1,poly(x,ord.model,raw=F),poly(z,ord.model,raw=F))
            form.data <- y~poly(x,ord.model,raw=F)+poly(z,ord.model,raw=F) }                            # Orthogonal
Polynomials
}

dat.sample=as.matrix(dat.sample)

## Generate parameter starting values based on user preferences
## Note: some of my choices here for starting value generation are extremely arbitrary.
##      The paper shows that the choice of starting values does not matter much.
if (H.true == H.model) {
if (inits == 0) { at.inits=matrix(0,S,H.true)
                           B.inits=matrix(0,H.true-1,p.true)
} else { at.inits=at.true[,1:H.true]+matrix(rnorm(S*H.true,0,abs(at.true/inits)),S,H.true)
              B.inits=B.true+matrix(rnorm((H.true-1)*p.true,0,abs(B.true/inits)),H.true-1,p.true) }
}

if (H.true != H.model) {
            at.inits=matrix(0,S,H.model)
            B.inits=matrix(0,H.model-1,p.true)
}

if (p.model < p.true) { B.inits=B.inits[,c(1,2:(ord.model+1),(ord.true+2):(ord.true+ord.model+1))] }
if (p.model > p.true) { B.inits=cbind(B.inits[,1:(ord.true+1)],matrix(0,nrow=H.model-1,ncol=ord.model-
ord.true),B.inits[,(ord.true+2):p.true],matrix(0,nrow=H.model-1,ncol=ord.model-ord.true)) }

## set up arrays that will contain true and progressively updated values for alpha/tau/mu
mu.true=invlogit(at.true[,1]+at.true[,2:(H.true+1)])
at.array=array(0,c(S,H.model+1,m+1))
at.array[,1:H.model,1]=at.inits
at.array[,H.model+1,1]=-apply(at.array[,2:H.model,1],1,sum)
at.update=at.array[,,1]
at.rcc=rep(0,m)

## set up arrays that will contain true and progressively updated values for B/pi
pi.true=t(apply(pmin(pmax(dat.sample.trueorder%*%t(B.true),-100),100),1,additive.logistic))
B.array=array(0,c(H.model-1,p.model,m+1))
B.update=B.array[,,1]=B.inits
B.rcc=rep(0,m)
loglikelihood=0

## Blocked Nonlinear Gauss-Seidel Algorithm
pb <- winProgressBar(title = "progress bar", min = 0,max = m, width = 300)
opt.start <- proc.time()
for (w in 1:m) {
            for (s in 1:S) {

            at.opt=optim(at.update[s,1:H.model],eval(as.name(paste("loglike.at",s,sep=''))),gr=NULL,at.update,B.updat
e,dat.sample,test$sample,method=method,control=list(fnscale=-1))
                        at.new.piece=at.opt$par
                        at.new.piece.h=c(at.new.piece,-sum(at.new.piece[2:H.model]))
```

```
                at.update[s,]=at.new.piece.h
        }
        at.array[,,w+1]=at.update
        at.rcc[w]=max(abs((at.array[,,w+1]-at.array[,,w])/(at.array[,,w]+epsilon)))          #relative convergence
criterion
        for (h in 1:(H.model-1)) {

        B.opt=optim(B.update[h,],eval(as.name(paste("loglike.B",h,sep=''))),gr=NULL,at.update,B.update,dat.sampl
e,test$sample,method=method,control=list(fnscale=-1,parscale=parscale))
                B.update[h,]=B.opt$par
        }
        B.array[,,w+1]=B.update
        B.rcc[w]=max(abs((B.array[,,w+1]-B.array[,,w])/(B.array[,,w]+epsilon))) #relative convergence criterion

        w.conv=w
        setWinProgressBar(pb, w, title=paste( round(w/m*100, 0),"% done"))
        print(paste("Completed Iterations:",w))
        if (at.rcc[w]<delta & B.rcc[w]<delta) {
                print(paste("ALGORITHM HAS CONVERGED AFTER",w,"ITERATIONS"))
                break()
        }
        if (w==m) {
                print(paste("ALGORITHM HAS NOT MET THE CONVERGENCE CRITERION AFTER",m,"ITERATIONS"))
        }
}
close(pb)
opt.stop <- proc.time()
opt.time=opt.stop-opt.start ; opt.time
n.iters <- w.conv ; n.iters

## Name estimated parameters and compute related pi and mu estimates.
at.est=at.array[,,w.conv+1]
mu.est=invlogit(at.est[,1]+at.est[,2:(H.model+1)])
B.est=B.array[,,w.conv+1]
pi.est=t(apply(dat.sample%*%t(B.est),1,additive.logistic))

################################################################################
#                                       OPTIMIZATION DIAGNOSTICS                                        #
################################################################################

at.rcc ; at.true ; at.est
B.rcc
mu.true ; mu.est

################################################################################
                ##  Correct Classification Rate and Habitat Swapping  ##
################################################################################

## See paper for discussion of and motivation for habitat swapping

rate=0
flip=backflip=1:H.model
```

```
if (H.true == H.model) {    # Swapping can only be done when this is true

votfun=function(x) {
  ismax=x==max(x)
  ((1:H.model)[ismax])[1] #uniqueness ensured
  }

## True Habitat Classifications
mytruedataprobs.joint=t(apply(pmin(pmax(dat.sample.trueorder%*%t(B.true),-100),100),1,additive.logistic))
truehabclass=apply(mytruedataprobs.joint,1,votfun) ; truehabclass

## Find all permutations of habitat labels
flips=permutations(H.model,H.model)
numflips=nrow(flips)

## Estimated Habitat Classifications
myestdataprobs.joint=t(apply(pmin(pmax(dat.sample%*%t(B.est),-100),100),1,additive.logistic))
rates=rep(0,numflips)
esthabclass=rep(0,n)
for (permutation in 1:numflips) {
   myestdataprobs=myestdataprobs.joint[,flips[permutation,]]
   esthabclass=apply(myestdataprobs,1,votfun)
   rates[permutation]=length(which((esthabclass-truehabclass)==0))/n
}
rates ; flips
rate=max(rates) ; rate
flip=flips[which.max(rates),] ; flip
backflip=(1:H.model)[order(flip)] ; backflip
}

################################################################################
                                    ##  Model Selection  ##
################################################################################

## Get LogLikelihood Value
Xdata=as.matrix(dat.sample)
Ydata=as.matrix(test$sample)
bin=array(NA,c(n,S,H.model))
for (j in 1:S) {
                logitmu=at.est[j,1]+at.est[j,2:(H.model+1)]
                mu=exp(logitmu)/(1+exp(logitmu))
                for (i in 1:H.model) { bin[,j,i]=ifelse(as.logical(Ydata[,j]),mu[i],1-mu[i]) }
}
b=matrix(NA,nrow=n,ncol=H.model)
for (k in 1:H.model) { b[,k]=apply(bin[,,k],1,prod.fun) }
expXB=matrix(NA,nrow=n,ncol=H.model-1)
for (l in 1:(H.model-1)) { expXB[,l]=exp(pmin(pmax(Xdata%*%cbind(B.est[l,]),-100),100)) }
M=matrix(NA,nrow=n,ncol=H.model)
for (mm in 1:(H.model-1)) { M[,mm]=expXB[,mm]/(1+apply(expXB,1,sum)) }
M[,H.model]=1-apply(M[,1:(H.model-1)],1,sum)
sitelike=apply(b*M,1,sum)
loglikelihood=sum(log(sitelike))
```

```
## AIC
numBparams=p.model*(H.model-1)
numATparams=S*(H.model+1)
k=numBparams+numATparams
AIC=2*k-2*loglikelihood

## AICc
N=n*S
AICc=AIC+(2*k*(k+1))/(N-k-1)

####################################################################################

## Print a bunch of results
rate
n ; S
H.true ; H.model
p.true ; p.model
m ; n.iters
AIC ; AICc
inits ; seed ; delta
method ; rawpoly ; parscale ; pixels
flip ; backflip
B.true ; B.inits ; B.est
opt.time

####################################################################################

## Save a bunch of results to savepath folder
if (saveplots) {
		write.table(round(rate,3),paste(newpath,"/~Success Rate.csv",sep=""))
		write.table(flip,paste(newpath,"/~flip.csv",sep=""))
		write.table(round(AIC,3),paste(newpath,"/~AIC.csv",sep=""))
		write.table(round(AICc,3),paste(newpath,"/~AICc.csv",sep=""))
		write.table(round(loglikelihood,3),paste(newpath,"/~Log Likelihood Value.csv",sep=""))
		write.table(n.iters,paste(newpath,"/~Number of Iterations.csv",sep=""))
		write.table(m,paste(newpath,"/~Maximum Iterations.csv",sep=""))
		write.table(round(opt.time[3],3),paste(newpath,"/~OptTime.csv",sep=""))
		write.table(inits,paste(newpath,"/~inits.csv",sep=""))
		write.table(seed,paste(newpath,"/~seed.csv",sep=""))
		write.table(tauseed,paste(newpath,"/~tauseed.csv",sep=""))
		write.table(delta,paste(newpath,"/~delta.csv",sep=""))
		write.table(B.true,paste(newpath,"/~B True.csv",sep=""))
		write.table(B.inits,paste(newpath,"/~B Starting Values.csv",sep=""))
		write.table(B.est,paste(newpath,"/~B MLE.csv",sep=""))
		write.table(parscale,paste(newpath,"/~parscale.csv",sep=""))
		write.table(H.true,paste(newpath,"/~H.true.csv",sep=""))
		write.table(H.model,paste(newpath,"/~H.model.csv",sep=""))
		write.table(p.true,paste(newpath,"/~p.true.csv",sep=""))
		write.table(p.model,paste(newpath,"/~p.model.csv",sep=""))
		write.table(n,paste(newpath,"/~n.csv",sep=""))
		write.table(S,paste(newpath,"/~S.csv",sep=""))
		save.image(file=paste(newpath,"/Workspace.RData",sep=""))
}
```

Appendix D – Plots and Diagnostics

```
################################################################################
```

This collection of code creates plots that evaluate the performance of parameter estimation. Specifically, plots directly comparing true to estimated parameter values, true and MLE habitat maps (individual and additive) and classification plots are generated.

If requested by the user, plots are output to the same save folder as described in the Parameter Estimation section.

The Testing Scenario Setup, Function Library and Parameter Estimation code sections must be run prior to this section.

```
################################################################################

## Allows toggling of various diagnostic/evaluation plots
param.estimation=T
mle.maps=T
pi.histogram=T
class=T


################################################################################
#                          Parameter Estimation & Convergence                        #
################################################################################

## Direct comparison of parameters to MLE estimates, and plots of relative
## convergence criteria.

if (param.estimation && H.true==H.model) {

windows(width=12,height=4.5)
par(mfcol=c(1,3),oma=c(0,0,2,0))
plot(c(at.true),c(at.est[,c(1,flip+1)]),xlab="True Value",ylab="MLE Value",main="Alpha/Tau")
abline(0,1)
plot(c(mu.true),c(mu.est[,flip]),xlab="True Value",ylab="MLE Value",main="MU")
abline(0,1)
plot(c(pi.true),c(pi.est[,flip]),xlab="True Value",ylab="MLE Value",main="PI")
abline(0,1)
title("Parameter & Pseudo-Parameter Estimation",outer=TRUE,line=-0.5,cex.main=2,font.main=4,col.main="blue")
if (saveplots) { dev.copy2pdf(file=paste(newpath,"/_Parameter Estimation",".pdf",sep="")) }

if (p.true == p.model) {
windows(width=8,height=4.8)
par(mfcol=c(1,2),oma=c(0,0,2,0))
at.rcc=at.rcc[which(at.rcc>0)]
plot(3:length(at.rcc),at.rcc[3:length(at.rcc)],type='l',main="Convergence of Alpha/Tau",xlab="iteration",ylab="RCC")
B.rcc=B.rcc[which(B.rcc>0)]
plot(3:length(B.rcc),B.rcc[3:length(B.rcc)],type='l',main="Convergence of B",xlab="iteration",ylab="RCC")
```

```
title("Algorithm Convergence",outer=TRUE,line=-0.5,cex.main=2,font.main=4,col.main="blue")
if (saveplots) { dev.copy2pdf(file=paste(newpath,"/_Algorithm Convergence",".pdf",sep="")) }
}

if (p.true == p.model) {
windows(width=4,height=4.8)
par(mfcol=c(1,1),oma=c(0,0,2,0))
B.true.0=rbind(B.true,0) ; B.est.0=rbind(B.est,0)
plot(c(B.true.0),c(B.est.0[flip,]),xlab="True Value",ylab="MLE Value",main="B")
abline(0,1)
title("B Estimation",outer=TRUE,line=-0.5,cex.main=2,font.main=4,col.main="blue")
if (saveplots) { dev.copy2pdf(file=paste(newpath,"/_B Estimation",".pdf",sep="")) }
}

}
################################################################################




################################################################################
#                                   Estimated Habitat Maps                                   #
################################################################################

if (mle.maps) {

## Recreate data (lat and long) at each point in a grid across the study region
xx=seq(-10,10,length.out=pixels) ; zz=seq(-10,10,length.out=pixels)
dat.grid <- data.frame(xx,zz)
dat.grid <- expand.grid(dat.grid$xx,dat.grid$zz)
dat.grid <- data.frame(y=1,dat.grid)                    ## create grid for map
names(dat.grid) <- c("y","xx","zz")
if (exists("rawpoly") && rawpoly) {
        form.data2 <- y~poly(xx,ord.model,raw=T)+poly(zz,ord.model,raw=T)
} else { form.data2 <- y~poly(xx,ord.model,raw=F)+poly(zz,ord.model,raw=F) }
X.model <- model.matrix(form.data2,dat.grid)

################################################################################
          ##  MLE Individual Logistic Habitat Probability Plots  ##

## Estimated 'Individual' logistic probabilities (see paper) for each point in grid
link.fun <- make.link("logit")
myprobs.indiv.mle=matrix(NA,nrow=pixels^2,ncol=H.model-1)
for (h in 1:(H.model-1)) {
        myprobs.indiv.mle[,h] <- link.fun$linkinv(pmin(pmax(X.model%*%B.est[h,],-100),100))
}
indivprobs.grid.mle <- array(NA,c(pixels,pixels,H.model-1))
for (i in 1:length(xx)) {
 for (j in 1:length(zz)) {
  for(k in 1:(H.model-1)) {
        indivprobs.grid.mle[i,j,k] <- myprobs.indiv.mle[,k][which(dat.grid$xx==xx[i] & dat.grid$zz==zz[j])]
  }
 }
```

```
}

## Define plot layout
if (H.model==2) { par(mfrow=c(1,2),oma=c(0,0,2,0))
} else if (H.model==3)  { windows(width=9,height=5) ; par(mfrow=c(1,2),oma=c(0,0,2,0))
} else if (H.model==4)  { windows(width=12,height=4.5) ; par(mfrow=c(1,3),oma=c(0,0,2,0))
} else if (H.model==5)  { windows(width=13,height=4) ; par(mfrow=c(1,4),oma=c(0,0,2,0))
} else if (H.model==6)  { windows(width=12,height=9) ; par(oma=c(0,0,2,0)) ;
layout(matrix(c(1,1,2,2,3,3,1,1,2,2,3,3,0,4,4,5,5,0,0,4,4,5,5,0), 4, 6, byrow=TRUE))
} else if (H.model==7)  { windows(width=12,height=9) ; par(mfrow=c(2,3),oma=c(0,0,2,0))
} else if (H.model==8)  { windows(width=13,height=7.5) ; par(oma=c(0,0,2,0)) ;
layout(matrix(c(1,1,2,2,3,3,4,4,1,1,2,2,3,3,4,4,0,5,5,6,6,7,7,0,0,5,5,6,6,7,7,0), 4, 8, byrow=TRUE))
} else if (H.model==9)  { windows(width=13,height=7.5) ; par(mfrow=c(2,4),oma=c(0,0,2,0))
} else if (H.model==10) { windows(width=12,height=15) ; par(mfrow=c(3,3),oma=c(0,0,2,0))
} else if (H.model==11) { windows(width=15,height=6.5) ; par(mfrow=c(2,5),oma=c(0,0,2,0))
} else if (H.model==12) { windows(width=13,height=10) ; par(oma=c(0,0,2,0)) ;
layout(matrix(c(1,1,2,2,3,3,4,4,1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,5,5,6,6,7,7,8,8,0,9,9,10,10,11,11,0,0,9,9,10,10,11,11,0),
6, 8, byrow=TRUE))
} else if (H.model==13) { windows(width=13,height=10) ; par(mfrow=c(3,4),oma=c(0,0,2,0))
} else { print("Matrix Layout not predifined for this number of Habitats") }

## Plot each habitat (& indicate where samples are taken)
for (l in 1:(H.model-1)) {
	image(xx,zz,indivprobs.grid.mle[,,l],xlab="Longitude",ylab="Latitude",main=paste("Habitat",l),zlim=c(0,1),c
ol=rev(rainbow(100,end=4/6)))
#	points(x,z,col="white",cex=4,pch='.')
}
title("MLE Individual Habitats",outer=T,line=-0.5,cex.main=2,font.main=4,col.main="blue")

if (saveplots) { dev.copy2pdf(file=paste(newpath,"/_MLE Individual Habitats",".pdf",sep="")) }

###############################################################################
		##  MLE Additive Logistic Habitat Probability Plots  ##

## Estimated 'Additive' logistic probabilities (see paper) for each point in grid
myprobs.joint.mle=t(apply(pmin(pmax(X.model%*%t(B.est),-100),100),1,additive.logistic))
jointprob.grid.mle <- array(NA,c(pixels,pixels,H.model))
for (i in 1:length(xx)) {
  for (j in 1:length(zz)) {
    for(k in 1:(H.model)) {
	jointprob.grid.mle[i,j,k] <- myprobs.joint.mle[,k][which(dat.grid$xx==xx[i] & dat.grid$zz==zz[j])]
    }
  }
}

jointprob.grid.mle=jointprob.grid.mle[,,flip]

## Define plot layout
if (H.model==2) { par(mfrow=c(1,2),oma=c(0,0,2,0))
} else if (H.model==3)  { windows(width=12,height=4.5) ; par(mfrow=c(1,3),oma=c(0,0,2,0))
} else if (H.model==4)  { windows(width=13,height=4) ; par(mfrow=c(1,4),oma=c(0,0,2,0))
} else if (H.model==5)  { windows(width=12,height=9) ; par(oma=c(0,0,2,0)) ;
layout(matrix(c(1,1,2,2,3,3,1,1,2,2,3,3,0,4,4,5,5,0,0,4,4,5,5,0), 4, 6, byrow=TRUE))
```

```
} else if (H.model==6)  { windows(width=12,height=9) ; par(mfrow=c(2,3),oma=c(0,0,2,0))
} else if (H.model==7)  { windows(width=13,height=7.5) ; par(oma=c(0,0,2,0)) ;
layout(matrix(c(1,1,2,2,3,3,4,4,1,1,2,2,3,3,4,4,0,5,5,6,6,7,7,0,0,5,5,6,6,7,7,0), 4, 8, byrow=TRUE))
} else if (H.model==8)  { windows(width=13,height=7.5) ; par(mfrow=c(2,4),oma=c(0,0,2,0))
} else if (H.model==9)  { windows(width=12,height=15) ; par(mfrow=c(3,3),oma=c(0,0,2,0))
} else if (H.model==10) { windows(width=15,height=7) ; par(mfrow=c(2,5),oma=c(0,0,2,0))
} else if (H.model==11) { windows(width=13,height=10) ; par(oma=c(0,0,2,0)) ;
layout(matrix(c(1,1,2,2,3,3,4,4,1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,5,5,6,6,7,7,8,8,0,9,9,10,10,11,11,0,0,9,9,10,10,11,11,0),
6, 8, byrow=TRUE))
} else if (H.model==12) { windows(width=13,height=10) ; par(mfrow=c(3,4),oma=c(0,0,2,0))
} else { print("Matrix Layout not predifined for this number of Habitats") }

## Plot each habitat (& indicate where samples are taken)
for (l in 1:(H.model)) {
		image(xx,zz,jointprob.grid.mle[,,l],xlab="Longitude",ylab="Latitude",main=paste("Habitat",l),zlim=c(0,1),col
=rev(rainbow(100,end=4/6)))
#		points(x,z,col="white",cex=4,pch='.')
}
title("MLE Additive Habitats",outer=T,line=-0.5,cex.main=2,font.main=4,col.main="blue")

if (saveplots) { dev.copy2pdf(file=paste(newpath,"/_MLE Additive Habitats",".pdf",sep="")) }

}

################################################################################



################################################################################
#                                   Reality Check                              #
################################################################################

if (pi.histogram && H.true==H.model) {
windows(width=7,height=5)
par(mfrow=c(1,1),oma=c(0,0,2,0))
diffs=c(abs(pi.true-pi.est[,flip]))
hist(diffs,xlab="Estimation Errors",main="PI:  |MLE-True|")
title("Reality Check: PIs",outer=TRUE,line=-0.5,cex.main=2,font.main=4,col.main="blue")
}
if (saveplots) { dev.copy2pdf(file=paste(newpath,"/_PI Histogram",".pdf",sep="")) }


## Check relation of B.est to B.true (boundary abruptness misestimation)
if (H.true==H.model) {
B.ratio=B.est/B.true
B.ratio=B.ratio[which(-1000<B.ratio & B.ratio<1000)]
B.scalar=round(sum(B.ratio)/length(B.ratio),2) ; B.scalar

B.est.adjust=B.est/B.scalar ; B.est.adjust
}


################################################################################
```

```
#                                   Classification Plots                                   #
###############################################################################

## Several versions of Classification plots, both for sampled sites and along
## a grid throughout the study region.  Many of these maps are provide redundant
## information, so they may be deleted.

if (class) {

###############################################################################
##                         Gridpoint Classification Plots                          ##
###############################################################################

windows(width=8,height=4.8)
par(mfcol=c(1,2),oma=c(0,0,2,0))

## Plot true and MLE classification plots side-by-side. (Important Evaluation Tool)
## true 'winner-takes-all' (classification) plot
plot(xx,zz,type="n",xlab="Longitude",ylab="Latitude",main="True")
points(dat.grid[,2],dat.grid[,3],col=colors[winner.true],pch=16)
xxx=rep(x,H.true)
zzz=rep(z,H.true)
#symbols(xxx,zzz,circles=sqrt(diffs),fg="white",bg="yellow",inches=0.1,add=T)

## estimated 'winner-takes-all' (classification plot
votfun=function(x) {
  ismax=x==max(x)
  ((backflip)[ismax])[1]       #uniqueness ensured
  }
winner.mle=apply(myprobs.joint.mle,1,votfun)
plot(xx,zz,type="n",xlab="Longitude",ylab="Latitude",main="MLE")
points(dat.grid[,2],dat.grid[,3],col=colors[winner.mle],pch=16)
xxx=rep(x,H.model)
zzz=rep(z,H.model)
if (H.true==H.model && p.true==p.model) {
        symbols(xxx,zzz,circles=sqrt(diffs),fg="white",bg="yellow",inches=0.1,add=T) }

title(expression("Classification Plots"),outer=TRUE,line=-0.5,cex.main=2,font.main=4,col.main="blue")

if (saveplots) { dev.copy2pdf(file=paste(newpath,"/_MLE Classifications",".pdf",sep="")) }

windows(width=4,height=4.4)
par(mfcol=c(1,1))

## estimated 'winner-takes-all' plot
votfun=function(x) {
  ismax=x==max(x)
  ((backflip)[ismax])[1]       #uniqueness ensured
  }
winner.mle=apply(myprobs.joint.mle,1,votfun)
plot(xx,zz,type="n",xlab="Longitude",ylab="Latitude",main="MLE")
points(dat.grid[,2],dat.grid[,3],col=colors[winner.mle],pch=16)
xxx=rep(x,H.model)
zzz=rep(z,H.model)
if (H.true==H.model && p.true==p.model) {
        symbols(xxx,zzz,circles=sqrt(diffs),fg="white",bg="yellow",inches=0.1,add=T) }

if (saveplots) { dev.copy2pdf(file=paste(newpath,"/_MLE Classification Plot (yellow dots)",".pdf",sep="")) }

windows(width=4,height=4.4)
par(mfcol=c(1,1))

## estimated 'winner-takes-all' plot
votfun=function(x) {
  ismax=x==max(x)
  ((backflip)[ismax])[1]        #uniqueness ensured
  }
winner.mle=apply(myprobs.joint.mle,1,votfun)
plot(xx,zz,type="n",xlab="Longitude",ylab="Latitude",main="MLE")
points(dat.grid[,2],dat.grid[,3],col=colors[winner.mle],pch=16)
xxx=rep(x,H.model)
zzz=rep(z,H.model)

if (saveplots) { dev.copy2pdf(file=paste(newpath,"/_MLE Classification Plot",".pdf",sep="")) }

##############################################################################
##                          Sample Classification Plots                     ##
##############################################################################

windows(width=8,height=4.8)
par(mfcol=c(1,2),oma=c(0,0,2,0))

## true sample classification plot
myprobs.true.dat=t(apply(pmin(pmax(dat.sample.trueorder%*%t(B.true),-100),100),1,additive.logistic))
votfun=function(x) {
  ismax=x==max(x)
  ((1:H.model)[ismax])[1]    #uniqueness ensured
  }
winner.true.dat=apply(myprobs.true.dat,1,votfun)
plot(xx,zz,type="n",xlab="Longitude",ylab="Latitude",main="True Classifications")
symbols(x,z,circles=rep(0.2,n),fg="black",bg=colors[winner.true.dat],inches=0.1,add=T)

## estimated sample classification plot
myprobs.est.dat=t(apply(pmin(pmax(dat.sample%*%t(B.est),-100),100),1,additive.logistic))
votfun=function(x) {
  ismax=x==max(x)
  ((backflip)[ismax])[1]        #uniqueness ensured
  }
winner.est.dat=apply(myprobs.est.dat,1,votfun)
plot(xx,zz,type="n",xlab="Longitude",ylab="Latitude",main="Estimated Classifications")
symbols(x,z,circles=rep(0.2,n),fg="black",bg=colors[winner.est.dat],inches=0.1,add=T)

title(expression("Sample Classification Plots"),outer=TRUE,line=-0.5,cex.main=2,font.main=4,col.main="blue")

if (saveplots) { dev.copy2pdf(file=paste(newpath,"/_Sample Habitat Classifications",".pdf",sep="")) }
}
```