Modelling Ecological Niche Using Lasso and Ridge Logistic Regression: A Study on *Mikania micrantha*

June 29, 2017

Modelling Ecological Niche Using Lasso and Ridge Logistic Regression: A Study on *Mikania micrantha*

Thesis submitted to the

INSTITUTE OF CHEMICAL TECHNOLOGY, MUMBAI

for the award of the degree of

MASTER OF SCIENCE

In the

ENGINEERING MATHEMATICS

by

SHOBHANA GOPAL IYER

under the supervision of

Dr. AMIYA R. BHOWMICK

Department of Mathematics

Institute of Chemical Technology, Mumbai

(University under Section 3 of UGC Act 1956;

Elite Status and Centre of Excellence, Government of Maharashtra)

Maharashtra, India

July 2017

DECLARATION BY THE CANDIDATE AS PER ORDINANCE

I hereby declare, as per ordinance ..... relating to the Degree (Master Of Science), that

(1) The thesis titled "**Modelling Ecological Niche Using Lasso and Ridge Logistic Regression: A Study on *Mikania micrantha***" submitted by me for the Degree (Master of Science) is the record of the research work carried out by me during the period from 2016 to 2016 under the guidance of my research guide **Dr. Amiya R. Bhowmick**.

(2) The work is original and whenever I have used materials (data, theoretical analyses, figures,text, etc.) from other sources, I have given due credit to them by citing them in the text of the thesis. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

(3) The work embodied in the thesis has not been submitted to this or any other University or Institute for the award of any degree, diploma, or certificate.

(4) I have followed the guidelines of the Institute in preparing the thesis. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute, including the policy of plagiarism.

(5) I hereby grant to the university and its agents the non-exclusive license to archive and make accessible, my thesis, in whole in all forms of media, now or hereafter known.

Shobhana Gopal Iyer

Dr. Amiya R. Bhowmick

CERTIFICATE OF RESEARCH GUIDE

This is to certify that the thesis "Modelling Ecological Niche Using Lasso and Ridge Logistic Regression: A Study on *Mikania micrantha*" submitted by Shobhana Gopal Iyer to the Institute of Chemical Technology, Mumbai, for the degree of "M.Sc in Engineering Mathematics" is a bonafide record of the research work carried out by her in the Department of Mathematics, Institute of Chemical Technology, Mumbai, under my supervision. *Shobhana Gopal Iyer* has worked under my guidance on this topic from January 2017 till June 2017.

1. The results embodied in this thesis have not been submitted to any other University or Institute (except for dual degree programme having MOU with (Name of university) for the award of any degree, diploma, or certificate.

2. The thesis has resulted in to number/no publications, number/no presentations and number/no patents which have been cited in this thesis and a few are likely to be published where a citation will be given to this thesis and it will be mentioned in the cover letter to the editor to avoid charges of plagiarism of my own work.

The thesis, in my opinion, is worthy of consideration for the award of the degree Doctor of Philosophy (Technology/Science/Pharmacy) in (subject) ..... in accordance with the Rules and Regulations of this University.

Date:

Dr. Amiya R Bhowmick

Acknowledgement

The completion of this study could not have been possible without the expertise of my guide Dr. A. R. Bhowmick. He has helped me reach my potential by being calm and polite every time. Thank you for the everlasting motivation and support you gave through out. Your positive outlook has given me the strength to do it all.

Also I express my deep sense of gratitude to Dr. Abhishek Mukherjee who despite his busy schedules, made time to visit our department and impart a lecture on *"Ecological niche modeling"*. His guidance, encouragement and suggestions have helped in the accomplishment of this project work.

I am thankful to Dr. Achyut Kumar Banerjee for providing the data on *Mikania micrantha* which was not available in GBIF. Several discussions with Dr. Banerjee helped us a lot to build this project with thorough understanding of the Niche modelling.

Ms. Shobhana Gopal Iyer

Contents

Abstract

Invasions by non-indigenous species are one of the major problem across the globe imposing large environmental impacts on the species in native range. Identifying the key environmental correlates favorable to rapid invasion speed is important for developing management strategies. In this project, we develop statistical model to identify the suitable environmental variables for the invasive plant species "*Mikania micrantha*". To build the model, the presence-only data was collected from Global Biodiversity Information Facility (GBIF). *Mikania* is native to South America. The occurrence data for India is collected from secondary sources. Four different backgrounds, namely South America, minimum convex polygon of South America, South America and India & minimum convex polygon of South America and India were used. The 19 bioclimatic variables were used from WorldClim to be further used as predictors in the model. The different modelling approaches are considered, viz logistic regression, ridge regression and lasso regression. Comparative performances of all these methods are elaborately discussed. At the end the suitable climate variables are identified and some future directions are given to improve the model further.

Chapter 1

Introduction

1.1 Spatial data

Spatial data, also known as geospatial data represents the geographic location of features or boundaries on earth. It can be represented as numerical values in geographic coordinate system. The coordinate reference system (CRS) provides a standardized way of representing locations. The geographic data can be described by using many different CRS like WGS84, NAD83 and NAD27. Amongst these CRS, WGS84 is the most commonly used as it provides geographic information system (GIS) data for entire globe. Spatial phenomena can be either discrete locations or continuous phenomenon with some boundaries. Discrete locations, assigned with some CRS are known as "spatial objects" which includes river, road, country etc. Continuous phenomenon like elevation, temperature, precipitation are called "spatial fields". Spatial objects' data describes the geometry of locations represented by vector data whereas spatial fields are represented by raster data.

1.1.1 Vector data

The main vector data types are points, lines and polygons. The geometry of these data structures consists of set of coordinate pairs (say (x,y)). Each point has a coordinate pair and some variables associated with it. The geometry of lines is represented as ordered sets of coordinates. Line segments can be drawn by connecting the points where ordering of point is important. The geometry of polygons is very similar to that of lines but the last coordinate pair must be the same as first coordinates pair.

Creating spatial objects in R

- `SpatialPoints` and `SpatialPointsDataFrame`: Spatial points are used to plot map of geospatial data whose coordinates refer to longitude and latitude information of the data. `SpatialPointsDataFrame` is a class for spatial attributes that have spatial point locations. For example, we may have information of some environmental covariates such as annual rainfall, annual mean temperature, etc. To store these informations at each spatial points, the `SpatialPointsDataFrame` is used.

```
> longitude <- c(-116.7, -120.4, -116.7, -113.5, -115.5,
-120.8, -119.5, -113.7, -113.7, -110.7)
> latitude <- c(45.3, 42.6, 38.9, 42.1, 35.7, 38.9, 36.2,
39, 41.6, 36.9)
> lonlat <- cbind(longitude, latitude)
> locations = as.data.frame(lonlat)
```

```
> SpPoints = SpatialPoints(lonlat)
> library(raster)
> crs(SpPoints) = '+proj=longlat +datum=WGS84'
> annual_rainfall = c( 17.9, 14.8,50, 5.5, 24.8, 20.05,
23.8, 25.4, 18.4, 12.3)
> D = data.frame(annual_rainfall)
> SpPointsdf = SpatialPointsDataFrame(SpPoints, D)
> SpPointsdf
class       : SpatialPointsDataFrame
features    : 10
extent      : -120.8, -110.7, 35.7, 45.3
(xmin, xmax, ymin, ymax)
coord. ref. : +proj=longlat +datum=WGS84
+ellps=WGS84 +towgs84=0,0,0
variables   : 1
names       : annual_rainfall
min values  :             5.5
max values  :              50
```

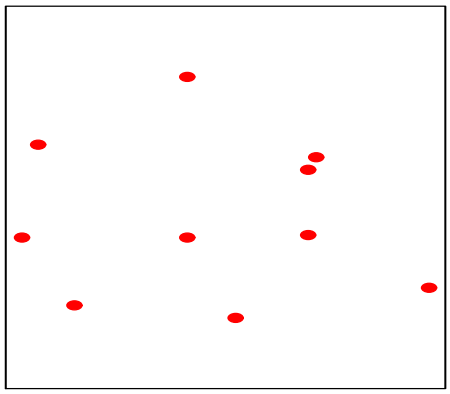


Figure 1.1: Vector points

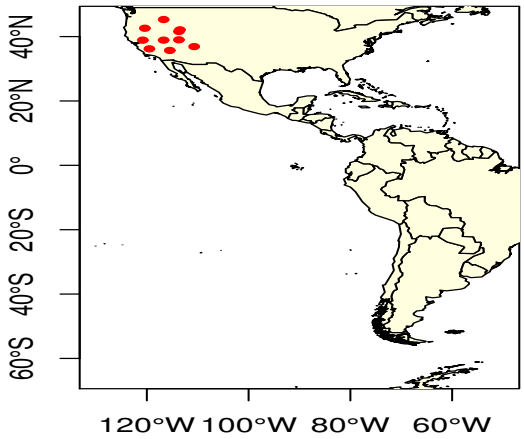


Figure 1.2: Spatial points plotted on world map

- `SpatialLines` and `SpatialLinesDataFrame`: Spatial lines are formed by joining the spatial points. It can be used to represent roads, rivers etc. in a map. The class `SpatialLinesDataFrame` holds the lines data that have a data frame attached to it.

```
> lon <- c(-116.8,-104.2,-92.9,-111.9,-114.2,-115.4,-121.7)
> lat <- c(41.3, 45.9, 42.4, 36.8, 37.6, 38.3, 37.6)
> lonlat <- cbind(lon, lat)
> lns <- spLines(lonlat, crs=crdref)
> L1 = Line(lonlat)
> L2 = Line(lonlat*2)
> Lines1 = Lines(list(L1,L2), ID = "a")
> Lines1 = Lines(list(L1,L1), ID = "a")
```

```
> Lines2 = Lines(list(L2,L2), ID = "b")
> crs(SpLines) = '+proj=longlat +datum=WGS84'
> SpLines
class       : SpatialLines
features    : 2
extent      : -243.4, -92.9, 36.8, 91.8
(xmin, xmax, ymin, ymax)
coord. ref. : +proj=longlat +datum=WGS84
+ellps=WGS84 +towgs84=0,0,0
```

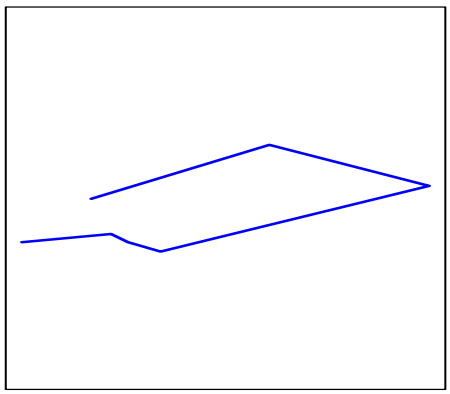


Figure 1.3: Lines plotted using the vector data

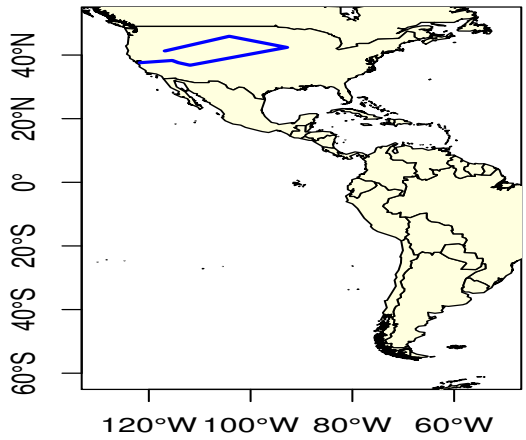


Figure 1.4: Spatial lines plotted on world map

- `SpatialPolygons` and `SpatialPolygonsDataFrame`: Spatial polygons can be drawn using spatial points. Polygons are closed hence, we need to join the first and last point. The class `SpatialPolygonsDataFrame`

are built from the SpatialPolygons object and the data frame.

```
> lon <- c(-116.8,-104.2,-92.9,-111.9,-114.2,-115.4,-116.8)
> lat <- c(41.3, 45.9, 42.4, 36.8, 37.6, 38.3, 41.3)
> lonlat <- cbind(lon, lat)
> Poly1 = Polygon(lonlat)
> Poly2 = Polygon(lonlat*2)
> P1 = Polygons(list(Poly1,Poly1), "p1")
> P2 = Polygons(list(Poly2,Poly2), "p2")
> SpPolygon = SpatialPolygons(list(P1,P2), 1:2)
> crs(SpPolygon) = '+proj=longlat +datum=WGS84'
> SpPolygon
class       : SpatialPolygons
features    : 2
extent      : -243.4, -92.9, 36.8, 91.8
(xmin, xmax, ymin, ymax)
coord. ref. : +proj=longlat +datum=WGS84
+ellps=WGS84 +towgs84=0,0,0
```

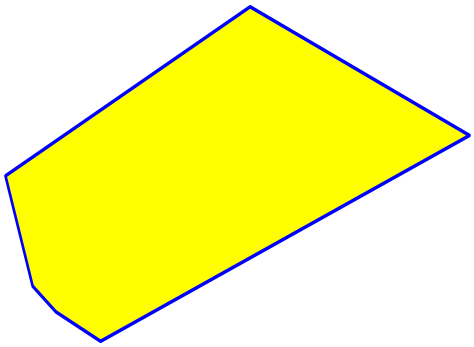


Figure 1.5: Polygon plotted using the vector data

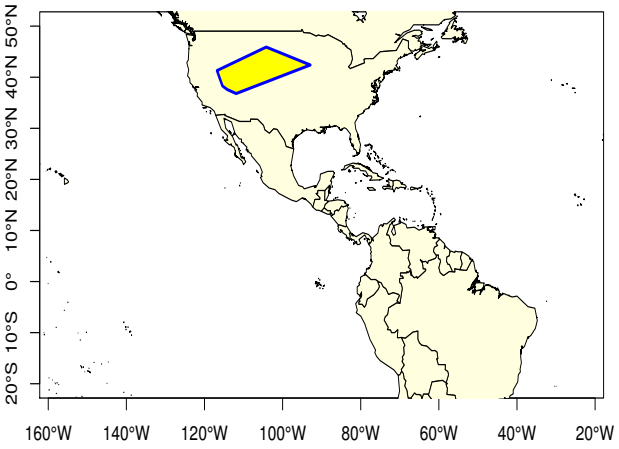


Figure 1.6: Spatial polygon plotted on world map

1.1.2 Raster data

Raster data is used to represent continuous variables. It consists of grid where each cell of the grid contains some value that represents information like temperature, precipitation, etc. Data stored in a raster format represents real-world phenomena like thematic data that represent land-use or soils data, continuous data that represents temperature, satellite images, aerial photographs, etc. Also raster data represents pictures like scanned maps, photographs etc. In Geographic information system(GIS), when comparing raster data with the other data types, resolution of raster dataset has to be understood. The resolution of raster determines the number of cells in rows and columns. In spatial data, resolution refers to the area covered on the geographical surface by a single cell. As the resolution of raster increases,

size of the cell decreases.

Creating spatial fields in R

The most important classes of `raster` package are `RasterLayer`, `RasterStack` and `RasterBrick`.

- `RasterLayer`: A `RasterLayer` object stores a number of parameters like number of rows and columns, spatial extent and CRS that describe it. It can also store the raster cell values in memory.

```
> bioclim_data_america <-system.file("C:/Users/admin/Desktop/
                                    bioclim_data_america.grd", package="r
> r = raster(bioclim_data_america, layer = 1)
```

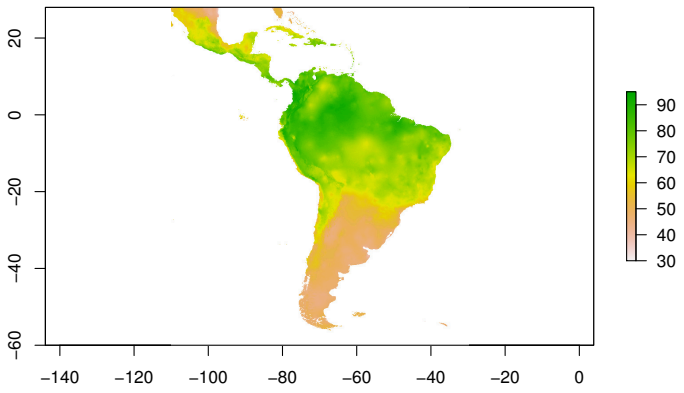


Figure 1.7: Raster Layer

- `RasterStack`: A `RasterStack` is a class for storing multi-layer data. It is a collection of `RasterLayer` objects with same spatial resolution and

extent. It can be formed from a collection of files in different locations or from few layers/bands from a file.

```
> bioclim_data_america <-
system.file("C:/Users/admin/Desktop/
bioclim_data_america.grd", package="raster")
> s2 = stack(s1, r)
> s2
class       : RasterStack
dimensions  : 2112, 1920, 4055040, 20
(nrow, ncol, ncell, nlayers)
resolution  : 0.04166667, 0.04166667  (x, y)
extent      : -110, -30, -60, 28
(xmin, xmax, ymin, ymax)
coord. ref. : +proj=longlat +datum=WGS84
+ellps=WGS84 +towgs84=0,0,0
names       :  bio1,  bio2, bio3.1,  bio4,
bio5,  bio6,  bio7,  bio8...
min values  :  -108,    18,     30,    95,
-21,  -217,    56,  -142...
max values  :   294,   211,     95,  7589,
411,   242,   356,   326...
```

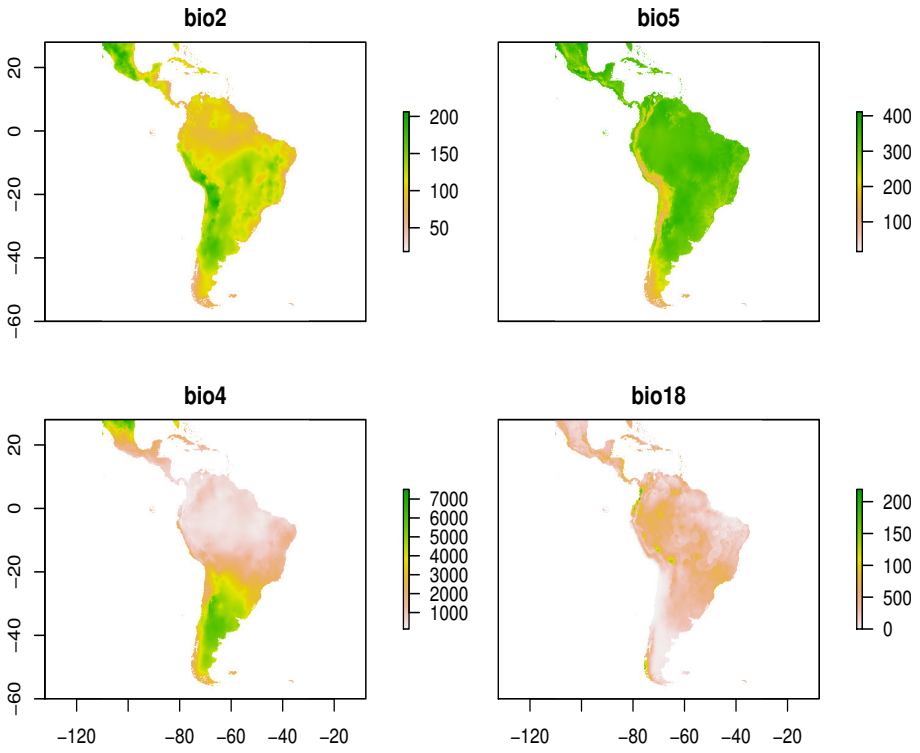


Figure 1.8: `RasterStack` created from a layer and a grid file

- `RasterBrick`: A `RasterBrick` is another class for storing multi-layer data. It can be linked only to a single(multi-layer) file. It is more efficient to process a `RasterBrick` than to process a `RasterStack`.

```
> bioclim_data_america <-
system.file("C:/Users/admin/Desktop/
bioclim_data_america.grd", package="raster")
> b = brick(bioclim_data_america)
> b
class       : RasterBrick
dimensions  : 2112, 1920, 4055040, 19
(nrow, ncol, ncell, nlayers)
```

```
resolution  : 0.04166667, 0.04166667
(x, y)
extent      : -110, -30, -60, 28
(xmin, xmax, ymin, ymax)
coord. ref. : +proj=longlat +datum=WGS84
+ellps=WGS84 +towgs84=0,0,0
```

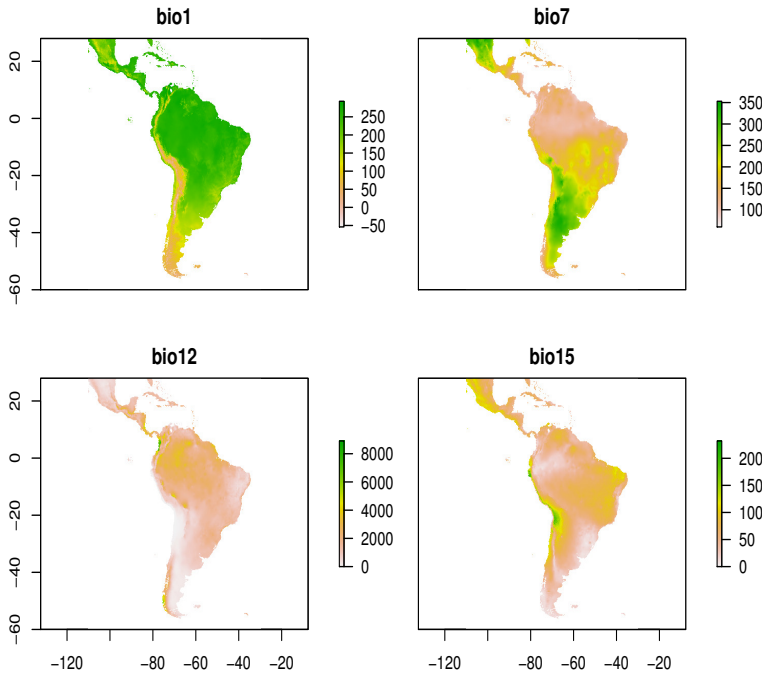


Figure 1.9: `RasterBrick` created from a single grid file

- `SpatialGrid`: The `SpatialGrid()` is used to define spatial grid by offset, cell size and dimensions. The class `GridTopology` is used for defining a complete rectangular grid of any dimension using the arguments `cellcenter`, `offset`, `cellsize` and `celldim`.

```
> grid = GridTopology(c(60,5), c(1,1), c(30,40))
> SpGrid = SpatialGrid(grid)
> crs(SpGrid) = '+proj=longlat +datum=WGS84'
> SpGrid = raster(SpGrid)
> values(SpGrid) = runif(1200)
> SpGrid
class       : RasterLayer
dimensions  : 40, 30, 1200
(nrow, ncol, ncell)
resolution  : 1, 1  (x, y)
extent      : 59.5, 89.5, 4.5, 44.5
(xmin, xmax, ymin, ymax)
coord. ref. : +proj=longlat +datum=WGS84
+ellps=WGS84 +towgs84=0,0,0
data source : in memory
names       : layer
values      : 0.002558375, 0.9987908
(min, max)
```

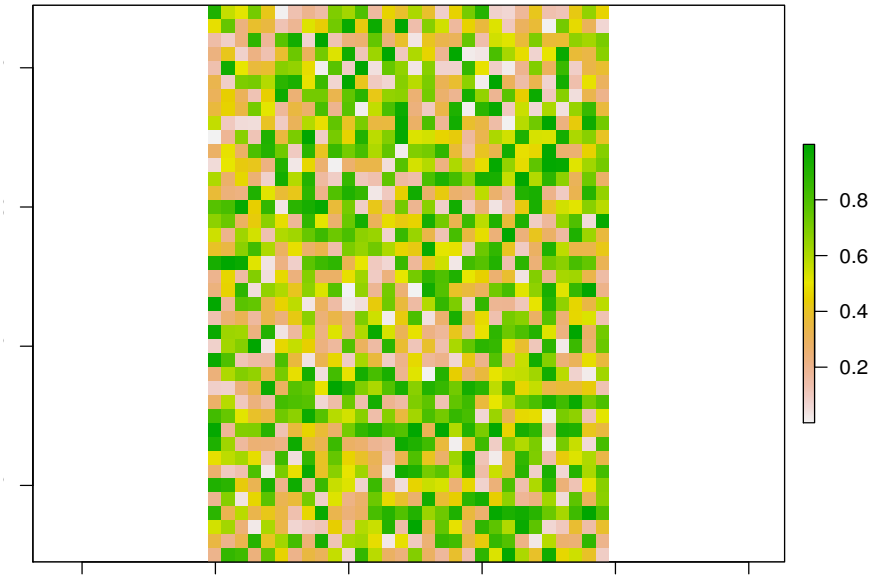


Figure 1.10: Spatial Grid

- `SpatialPixel`: The `SpatialPixel` defines pixel for incomplete rectangular grid of arbitrary dimension. It stores the grid topology and coordinates of the actual points which may be the subset of full grid. `SpatialPixelsDataFrame` consists of spatial attributes that have spatial locations for the grid topology instead of full rectangular grid.

```
> df = data.frame(z = c(1:6,NA,8,9),
+                 xc = c(1,1,1,2,2,2,3,3,3),
+                 yc = c(rep(c(0, 1.5, 3),3)))
> coordinates(df) = ~xc+yc
> gridded(df) = TRUE
> image(df["z"])
> df = data.frame(z = c(1:6,NA,8,9),
```

```
+                   xc = c(1,1,1,2,2,2,3,3,3),
+                   yc = c(rep(c(0, 1.5, 3),3)))
> coordinates(df) = ~xc+yc
> gridded(df) = TRUE
> image(df["z"])
> df
Object of class SpatialPixelsDataFrame
Object of class SpatialPixels
Grid topology:
cellcentre.offset cellsize cells.dim
xc                 1      1.0         3
yc                 0      1.5         3
SpatialPoints:
xc  yc
1  1 0.0
2  1 1.5
3  1 3.0
4  2 0.0
5  2 1.5
6  2 3.0
7  3 0.0
8  3 1.5
9  3 3.0
Coordinate Reference System (CRS) arguments: NA
```

```
Data summary:
z
Min.   :1.00
1st Qu.:2.75
Median :4.50
Mean   :4.75
3rd Qu.:6.50
Max.   :9.00
NA's   :1
```

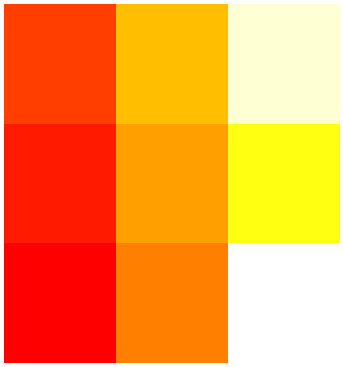


Figure 1.11: Spatial Pixel

Remark 1. *In R, raster data can be dealt using* `SpatialGridDataFrame` *and* `SpatialPixelsDataFrame` *from* `sp` *package and* `RasterLayer`, `RasterBrick` *and* `RasterStack` *from* `raster` *package. In this project the functions from*

`raster` package are used.

Chapter 2

General literature search including theoretical background

2.1 Data preparation for ecological niche modelling

2.1.1 Getting occurrence data from open source

The species data was accessed from the database, Global Biodiversity Information Facility (GBIF) (http://www.gbif.org/). The longitude-latitude information of the locations where the species occur was downloaded in R using `gbif()` function. The following code can be used to retrieve the data from `GBIF`.

```
> library(dismo)
> mic_gbif = gbif (genus = "Mikania",
species="micrantha Kunth", geo = TRUE)
> mic_gbif = mic_gbif[, c("species", "lon", "lat")]
> dim(mic_gbif)
```

```
[1] 3703    3
```

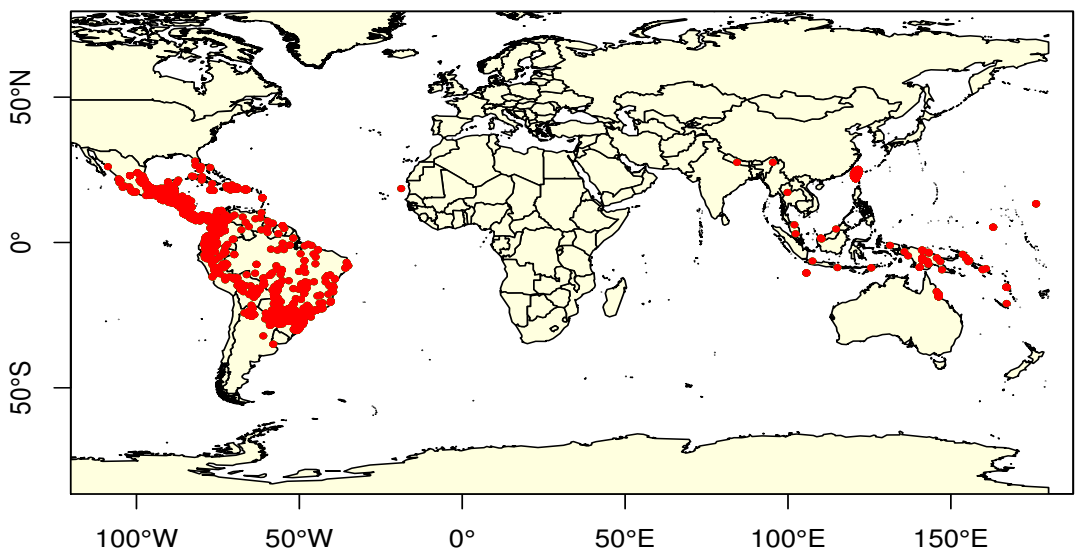


Figure 2.1: Occurrence location of Mikania micrantha Kunth

It was observed that the species was most likely to occur in the region of South America and then in some parts of India. Since we are concerned with these two regions further, we download shape files for these two regions in `SpatialPolygonsDataFrame` format from *GADM*(Database of Global Administrative Areas).

Figure 2.2: The region of South America and India in `SpatialPolygonsDataFrame`

2.1.2 Getting environmental covariates from WorldClim

We have also downloaded the bio-climatic variables from *WorldClim* which consists of information about the temperature, rainfall and precipitation of the whole world. It consists of 19 environmental covariates in the form of raster data. Each covariate forms a layer and all of these layers can be downloaded in the form of `RasterStack` in R using `getData()`. The environmental covariates can be masked for the region of South America and India using `mask()`.

```
> bioclim_data=getData(name="worldclim", download=TRUE,
var="bio", res=2.5, path="C:/Users/admin/Desktop")
```

```
> class(bioclim_data)
[1] "RasterStack"
attr(,"package")
[1] "raster"
> bioclim_data_america = mask(bioclim_data, south_america)
> bioclim_data_india = mask(bioclim_data_india, india)
```

2.1.3 Data cleaning

Removing duplicates

Now, we remove the duplicates with respect to longitude and latitude.

```
> dups = duplicated(mic_gbif_america[,c('lon', 'lat')])
> mic_gbif_america = mic_gbif_america[!dups,]
> write.csv(mic_gbif_america,          "mic_gbif_america.csv",
row.names = FALSE)
```

Removing points falling in ocean

In order to identify the coordinates which are falling in the ocean, first we attach the coordinate reference system with mic_gbif_america by converting it into `SpatialPointsDataFrame`. The `over()` retrieves the locations of mic_gbif_america from wrld_simpl and stores in `ovr`. Now in a new vector `country` the retrieved country from `ovr` is stored. If some longitude-latitude are in ocean then that will not be a part of any country. Now we save the

row numbers that correspond to none of the countries. Obtain the longitude and latitude of those rows and delete them from the coordinates of mic_gbif_america.

```
> library(sp)
> coordinates(mic_gbif_america) = ~lon+lat
> class(mic_gbif_america)
> crs(mic_gbif_america ) = crs(wrld_simpl)
> ovr = over(mic_gbif_america, wrld_simpl)
> country = ovr$NAME; sum(is.na(country))
> is_ocean = which(is.na(country))
> lonlat_ocean = mic_gbif_america@coords[is_ocean,]
> mic_gbif_america@coords = mic_gbif_america@coords[-is_ocean,]
```

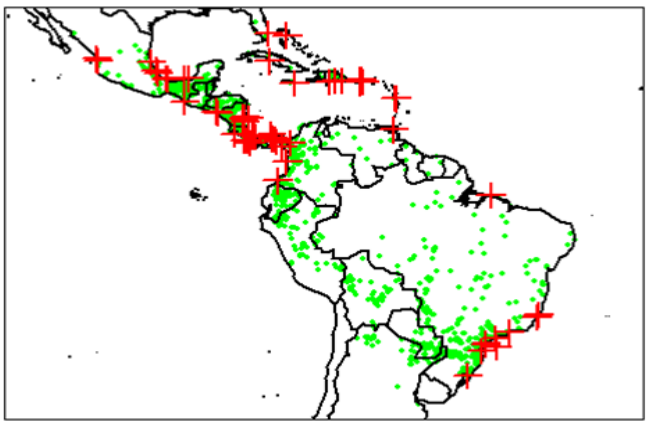


Figure 2.3: The occurrence points are marked with green color and the points falling in ocean with red color

2.1.4 Geographic bias

A critical issue while dealing with the presence-only data is the presence of sample selection bias. Occurrence data and climatic data are frequently biased due to transcription errors, lack of sufficient geographic details, data manipulation or low-resolution climatic data. It is often observed that the investigator visit areas for sample collection which are either geographically convenient or environmental-friendly. This gives rise to two different categories of selection bias, one occurs in the geographic space and the other in environmental space.

The geographical bias can be removed by dividing the whole region into grids and select one occurrence points from each grid Kadmon and Danin (2004)

```
> r = raster(mic_gbif_america)
> res(r) = 0.5
> r = extend(r, extent(r)+0.2)
> occ_one = gridSample(mic_gbif_america, r, n=1)
> dim(occ_one)
> net = rasterToPolygons(r)
> plot(net, border = 'gray')
> points(mic_gbif_america)
> points(occ_one, cex = 1, col ='red',pch = 'x')
> write.csv(occ_one, "mic_gbif_america.csv", row.names = FALSE)
```

Further we require environmental covariates at the occurrence locations that we have after cleaning the data. Using `extract()` we extract the environmental covariates corresponding to these locations.

```
> predictors_values_america = extract(bioclim_data_america,
mic_gbif_america[, c("lon", "lat")])
```

`predictors_values_america` contains the climatic information only in those locations where the species is observed in the region of South America. It might happen that, at those coordinates the climatic variables may not be available Those locations can not be used for modeling, hence we need to remove them from `predictors_values_america`.

```
> ind=complete.cases(as.data.frame(predictors_values_america))
> predictors_values_america =
as.data.frame(predictors_values_america[ind,])
```

Now we have the climatic information corresponding to the occurrence locations in the region of South America.

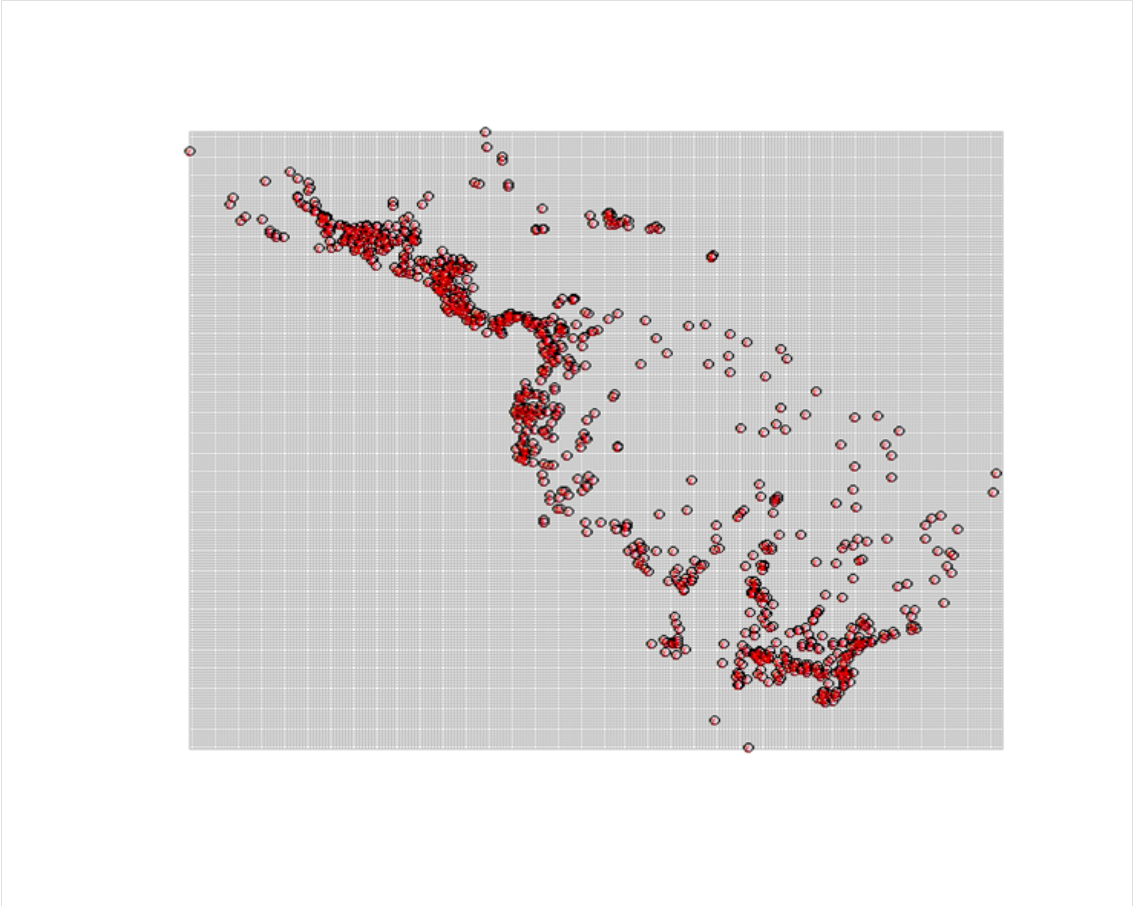


Figure 2.4: The fishnet with equal grid size of resolution 0.5. Only one occurrence point is sampled from each grid. The others are removed to avoid geographic bias in the data.

2.1.5 Environmental bias

We capture the background environment information and create the complete environmental space of South America. If the environmental covariates at occurrence records appear to represent a specific segment or clusters in the complete environmental space, then we suspect for the plausible presence of environmental bias in the occurrence data.

We randomly select a large number of locations from the complete background of South America. The function `randomPoints()` returns a set of

randomly selected locations. We select 10000 background locations and retrieve the quantitative values of environmental variables at those using the `extract()` function. Remove the missing values (if any) after extracting the environmental covariates information.

```
> library(dismo)
> lonlat_america = randomPoints(bioclim_data_america, 10000)
> bioclim_values_america =
extract(bioclim_data_america, lonlat_america)
> sum(complete.cases(as.data.frame( bioclim_values_america)))
> ind
=complete.cases(as.data.frame(bioclim_values_occ_america))
> bioclim_values_occ_america = bioclim_values_occ_america[ind,]
> par(mfrow = c(2,2))
> plot(bioclim_values_america[,1], bioclim_values_america[,6],
col="green", pch=16, xlab="Annual Mean Temperature(C)",
ylab = "Min Temp of Coldest Month (C)")
> points(bioclim_values_occ_america[,1],
> bioclim_values_occ_america[,6], col="red", pch=16)
> legend("topleft", legend = c("background", "occurrence"),
col = c("green", "red"), bty = "n", pch = c(16,16))
> plot(bioclim_values_america[,4], bioclim_values_america[,6],
col="green", pch=16,
xlab ="Temperature Seasonality(sd*100)",
```

```
ylab = "Min Temp of Coldest Month (C)")
> points(bioclim_values_occ_america[,4],
bioclim_values_occ_america[,6], col="red", pch=16)
> legend("topright", legend = c("background ", "occurrence "),
col = c("green", "red"), bty = "n", pch = c(16,16))
> plot(bioclim_values_america[,7], bioclim_values_america[,12],
col="green", pch=16,
xlab = "Temperature Annual Range (BIO5-BIO6) (C)",
ylab = "Annual Precipitation")
> points(bioclim_values_occ_america[,7],
bioclim_values_occ_america[,12], col="red", pch=16)
> legend("topright", legend = c("background ", "occurrence "),
col = c("green", "red"), bty = "n", pch = c(16,16))
> plot(bioclim_values_america[,12],bioclim_values_america[,15],
col="green", pch=16,
xlab = "Total(annual) precipitation",
ylab = " Precipitation seasonality (cv)")
> points(bioclim_values_occ_america[,12],
bioclim_values_occ_america[,15], col="red", pch=16)
> legend("topright", legend = c("background ", "occurrence "),
col = c("green", "red"), bty = "n", pch = c(16,16))
```

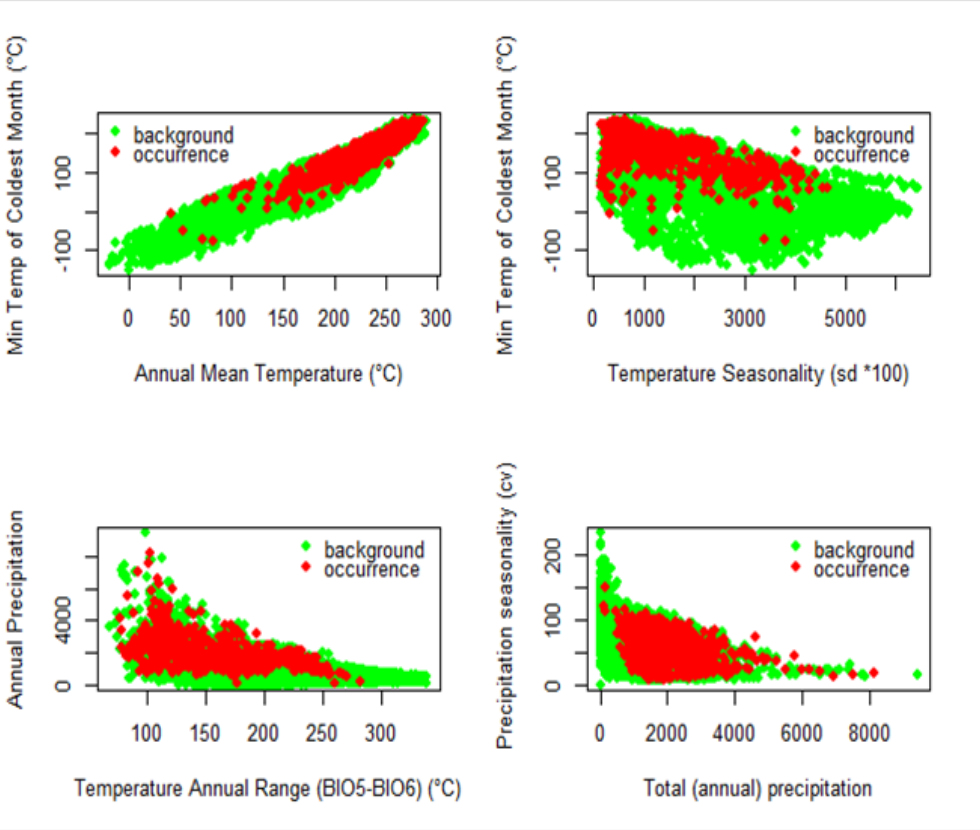


Figure 2.5: The plot of "Minimum temperature of coldest month" against "Annual mean temperature" & the plot of "Minimum temperature of coldest month" against "Temperature seasonality" has environmental bias. The red dot corresponds to the occurrence points and the green dots represents the background. If the red dots forms a cluster in a particular region covered by green dots, then one may suspect the environmental bias in the data.

2.1.6 Minimum convex polygon

We construct a minimum convex polygon that covers all the occurrence points in South America.

```
> library(adehabitatHR)
> Convex_poly = mcp(mic_gbif_america, percent = 100)
```

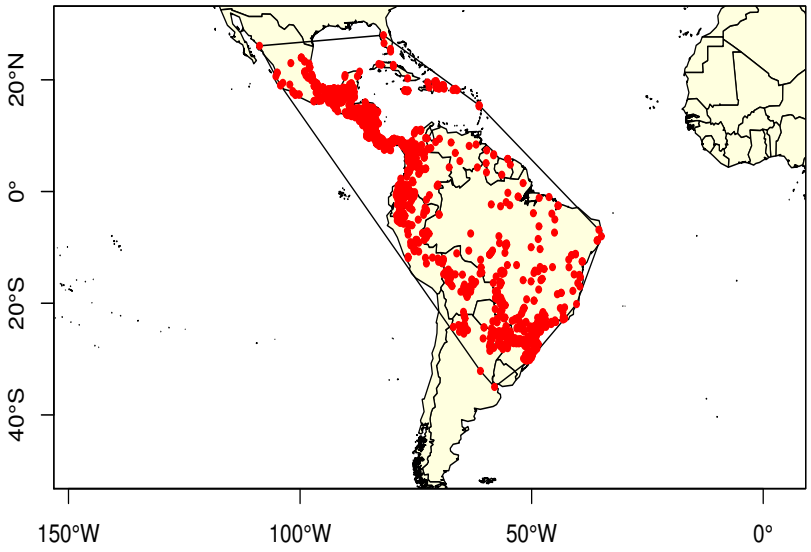


Figure 2.6: A minimum convex polygon is built that covers all the occurrence points in South America.

Now we obtain the region of South America that is covered in this minimum convex polygon by using `GIntersection()` from `rgeos` package. As per our observation some of the points lie on the boundary of the polygon, hence we buffer the polygon with some specified width so that all points lie inside the boundary.

```
> library(rgeos)
> Min_convex_poly = gIntersection(wrld_simpl, Convex_poly)
> Min_convex_poly_extended = gBuffer(Min_convex_poly,
byid=FALSE, id=NULL, width=1.0, quadsegs=5,
capStyle="ROUND", joinStyle="ROUND", mitreLimit=1.0)
```

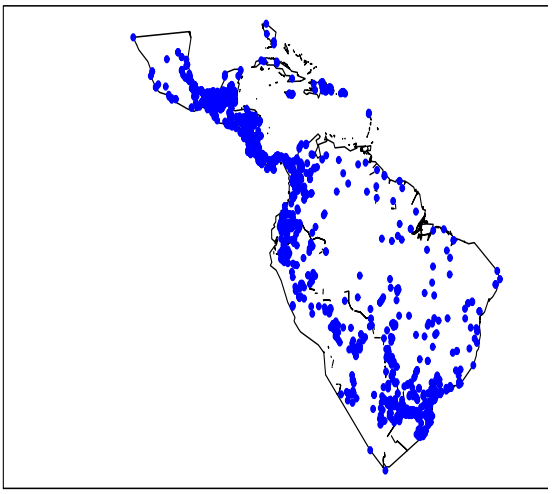


Figure 2.7: Before buffering some of the points lies on the boundary.

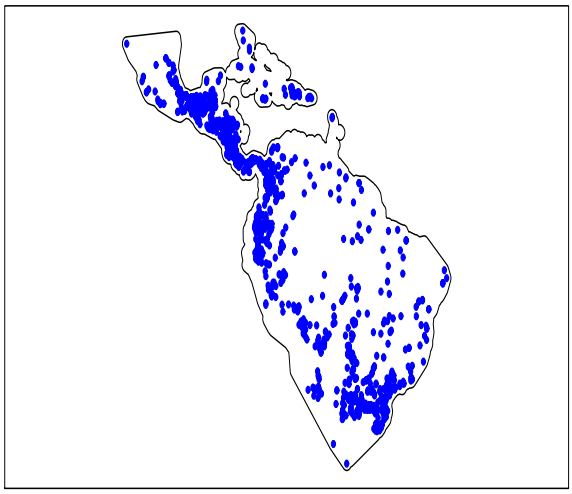


Figure 2.8: After buffering all the points lies inside the boundary.

We now choose the background points from this region and extract the environmental covariates for this region using `extract()`. We compare the results that we will be obtaining by selecting background points from the whole region of South America as well as from the minimum convex polygon that we have obtained just now.

2.2 Model selection and regularization

In this section, we distinguish between logistic regression, ridge regression and lasso regression. We discuss their comparative advantages and disadvantages

in real data analysis.

2.2.1 Linear regression

We start our discussion by the concepts of linear regression. The similar notations would be continued till the end of the manuscript while discussing the simple logistic, ridge logistic and least absolute shrinkage and selection operator (lasso) logistic regression techniques. We denote the response variable by Y and the set of predictor variables by $X_1, X_2, \cdots, X_p$, p being the number of predictors which are also synonymously written as explanatory variables, independent variables, covariates, regressors etc. The true relationship between between the response and the predictors can be approximated by a regression function f, so that the

$$Y = f(X_1, X_2, \cdots, X_p) + \epsilon \tag{2.1}$$

is a valid statistical model for the population of interest. Usually, f is a fixed but unknown function of $X_1, X_2, \cdots, X_p$ which represents the systematic variation of Y explained by the predictors. The unexplained component, denoted by ϵ, is assumed to be a random error (independent of the predictors) with mean zero. This gives a measure of discrepancy of the approximation by the function f James et al. (2014).

Under the multiple regression set up, f is replaced by a linear function of the predictors, represented as $Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \epsilon$, where β_0 is the intercept term and β_j gives the contributions of X_j for $j = 1, \cdots, n$ in explaining the variation of Y. For convenience, we adopt the matrix notation and represent the data set up. We have the continuous response $\mathbf{Y} = (y_1, \cdots, y_n)' \in \mathbb{R}^n$

and the n data values $(x_{ij})_{i=1}^{n}$ are available on each X_j, $j = 1, 2, \cdots, p$. The regression equation in matrix form written as

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{e} \tag{2.2}$$

where $\mathbf{X}$ is an $n \times (p+1)$ design matrix with entries in the first column being 1 (if intercept included) otherwise it is $n \times p$ order matrix with x_{ij} denotes the ith observation corresponding to the jth variable. $\boldsymbol{\beta}$ be the vector of regression coefficients of order $(p+1) \times 1$ and $\mathbf{e} = (e_1, e_2, \cdots, e_n)' \in \mathbb{R}^n$ is the vector of errors. By using the least squares theory, the residual sum of squares, $\mathbf{e}'\mathbf{e} = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij}\right)^2$ $(\mathbf{RSS}(\boldsymbol{\beta}))$, is minimized with respect to $\boldsymbol{\beta}$. Solving the normal equations, the estimates of $\boldsymbol{\beta}$ is obtained as $\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$. Using the notion of $l_2 - norm$, one can write as

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}} \left(\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2\right) \tag{2.3}$$

where $\|\mathbf{u}\|_2^2 = \sum_{i=1}^{n} u_i^2$ for a vector $\mathbf{u} \in \mathbb{R}^n$. $\hat{\boldsymbol{\beta}}$ is an unbiased and consistent estimator of $\boldsymbol{\beta}$ (Montgomery et al., 2006). The normal equations for the above minimization problem is given by $(\mathbf{X}'\mathbf{X})\boldsymbol{\beta} = \mathbf{X}'\mathbf{Y}$.

2.2.2 Multi-collinearity

In the previous section, if the matrix $(\mathbf{X}'\mathbf{X})$ has determinant zero, then the unique solution for the system of normal equations can not be obtained. If the matrix is ill-conditioned that is the determinant is very small, then variance for estimated coefficients are very large making the estimates unreliable (Hines et al., 2003). This is a common case in many real life data sets where

high degree of correlation exits between two variables. If any particular predictor can be closely approximated by a linear combination of two or more other predictors, then also the matrix $(\mathbf{X}'\mathbf{X})$ is nearly singular. Such a situation is called multicollinearity and must be taken care of before any modeling assignment (Graham (2003)).

2.2.3 Ridge regression

To deal with this various shrinkage methods have been proposed in the literature (Reference). The ridge regression technique, proposed by Hoerl and Kennard (1970a,b), deals with solving the normal equations of the form $(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})\,\boldsymbol{\beta} = \mathbf{X}'\mathbf{Y}$, where $\lambda \geq 0$ is called the shrinkage parameter. Because of the additional parameter λ the coefficient estimates $\hat{\boldsymbol{\beta}}$ has lower variance but they are no longer unbiased (Hines et al., 2003). Essentially, instead of minimizing the residual sum of squares, ridge regression minimizes a slightly different quantity, given by $\mathbf{RSS}(\boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} \beta_j^2$. The coefficient estimates can be written using $l_2 - norm$ as

$$\hat{\boldsymbol{\beta}}_{\lambda}^{R} = \text{argmin}_{\boldsymbol{\beta}} \left(\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_2^2 \right) \tag{2.4}$$

The term $\lambda \sum_{j=1}^{p} \beta_j^2$, called the shrinkage penalty, is small when $\beta_1, \cdots, \beta_p$ are close to zero. For $\lambda = 0$, the procedure is equivalent to ordinary least squares regression. For $\lambda \to \infty$, the impact of shrinkage penalty grows and the components of $\hat{\boldsymbol{\beta}}_{\lambda}^{R}$ will approach zero and its successful application is guaranteed by the bias-variance trade-off (James et al., 2014).

2.2.4 Lasso regression

The model selection methods such as forward, backward and mixed selection give the model that involve a subset of all the predictors. These techniques can be conveniently performed using R software for statistical computing (R Core Team, 2016). The packages `ISLR, leaps, MASS` can be used for this purpose and different criteria such as R^2, AIC etc. can be utilized for the model selection. Unlike these methods, ridge regression includes all p predictors in the model; the penalty term will make many coefficients very small, but will not set them exactly equal to zero (James et al., 2014). This is challenging for interpretation of the results in high dimension when the number of predictors is very large. This short coming is overcome by the method of lasso regression by minimizing the quantity $\mathbf{RSS}(\boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} |\beta_j|$, which considers an l_1-norm penalty instead of l_2-norm penalty (Tibshirani, 1996; Knight and Fu, 2000). Thus using $l_1 - norm$ notation, the coefficient estimates can be written as

$$\hat{\boldsymbol{\beta}}_{\lambda}^{L} = \text{argmin}_{\boldsymbol{\beta}} \left(\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1 \right) \tag{2.5}$$

where $\|\mathbf{u}\|_1 = \sum_{i=1}^{n} |u_i|$ for a vector $\mathbf{u} \in \mathbb{R}^n$. Because of the l_1 penalty some of the coefficients are forced to be equal to zero for large λ. Thus like the best subset selection methods, lasso also provides variable selection method. More precisely the lasso regression falls between best subset selection method and the ridge regression method and has some nice statistical properties from both techniques (Hastie et al., 2009).

The above discussion provided some preliminary ideas on how the devel-

opment from ordinary least square regression to the lasso regression were developed under a continuous regression set up. Recall from the Sect. 2.1, we have presence-only data for the *Mikania micrantha* species from South America and India. After selecting the pseudo-absence from the background locations, we populate the response variable $\mathbf{Y}$ with 1 (for presence) and 0 (for background). In each presence and background locations (defined by longitude and latitude) the values of environmental covariates are available. Thus our problem for predicting the response at new locations falls under the classification problem. Continuing the same notations (as in continuous response set up), we discuss the logistic regression method with ridge and lasso type penalty.

2.2.5 Logistic regression

The logistic regression is the model for $\pi(\mathbf{x}) = \mathbb{P}(Y = 1|X = \mathbf{x})$ at the predictor values $\mathbf{x} = (x_1, \cdots, x_p)$ for p predictors and is given by

$$\text{logit}[\pi(\mathbf{x})] = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p \tag{2.6}$$

which directly simplifies to

$$\pi(\mathbf{x}) = \frac{\exp\left(\beta_0 + \sum_{i=1}^{p} \beta_i x_i\right)}{1 + \exp\left(\beta_0 + \sum_{i=1}^{p} \beta_i x_i\right)} \tag{2.7}$$

Here the values y_i of the response variable are either 0 or 1. Let $\mathbf{x_i}$ denotes the ith row of the design matrix $\mathbf{X}$. To estimate the parameters for the logistic regression model instead of minimizing the sum of squares the maximum likelihood estimation method is used in which the log-likelihood function is maximized numerically. The joint probability mass function is proportional

to

$$\prod_{i=1}^{n} \pi(\mathbf{x_i})^{y_i}[1-\pi(\mathbf{x_i})]^{1-y_i}.$$

So the log-likelihood function is given by

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{n} [y_i \log(\pi(\mathbf{x_i})) + (1-y_i)\log(1-\pi(\mathbf{x_i}))] \tag{2.8}$$

Replacing $\pi(\mathbf{x_i})$, the final log-likelihood function is of the following form

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left[y_i \left(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} \right) - \log \left(1 + \exp(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}) \right) \right]. \tag{2.9}$$

Now $\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}} \{l(\boldsymbol{\beta})\}$ is the estimate of $\boldsymbol{\beta}$; the associated numerical procedure by Newton-Raphson method and the asymptotic properties are discussed in (Agresti, 2007). If there exists multicollinearity in the predictors the problems remains same as discussed earlier. Adding $l_2 - norm$ penalty term to the likelihood function, ridge logistic regression was developed and the log-likelihood function is written as (le Cessie and van Houwelingen, 1992),

$$l_\lambda^R(\boldsymbol{\beta}) = l(\boldsymbol{\beta}) - \lambda \|\boldsymbol{\beta}\|_2^2 \tag{2.10}$$

The lasso logistic regression considers the l_1-norm penalty in the log-likelihood function which is given by (Tibshirani, 1996)

$$l_\lambda^L(\boldsymbol{\beta}) = l(\boldsymbol{\beta}) - \lambda \|\boldsymbol{\beta}\|_1 \tag{2.11}$$

Under binary response set up, $\hat{\boldsymbol{\beta}}_\lambda^R = \text{argmax}_{\boldsymbol{\beta}} \left\{ l_\lambda^R(\boldsymbol{\beta}) \right\}$ and $\hat{\boldsymbol{\beta}}_\lambda^L = \text{argmax}_{\boldsymbol{\beta}} \left\{ l_\lambda^L(\boldsymbol{\beta}) \right\}$. The qualitative behavior of the outputs in terms of variable selection for different choices of λ on $[0, \infty)$ is similar as discussed for continuous set up. Zou and Hastie (2003) proposed a modified regularization and model selection method when combination of both l_1 and l_2 type penalties is used as

$\alpha\|\boldsymbol{\beta}\|_2^2 + (1-\alpha)\|\boldsymbol{\beta}\|_1$ with $\alpha \geq 0$ being the tuning parameter. This approach is called 'elastic net' and in particularly useful in high dimension ($n < p$).

2.2.6 Choice of the tuning parameter λ

Different choice of λ in the likelihood functions (2.10) and (2.11) will produce different coefficient estimates. Choosing an optimal value of λ is crucial to apply such shrinkage methods successfully. One possible way to consider train-test validation, in which the available set of observations are divided into two parts (for example 80% and 20%). One set is called the training set (usually the bigger set) and the other set is called the validation set or hold-out set. The model is fitted using the data from training set and is the used to predict the response in the validation set. The process is repeated and the average test error is computed for different choice of λ. The value of λ is chosen as optimal that gives the lowest test error rate. Under continuous regression set up usually the average test mean square error (mse) is computed and for classification set up average test misclassification error rate is computed. In this article, to choose λ, along with the train-test validation, we shall also use k-fold cross validation where instead of partitioning the data into two parts, the full data is divided into k non-overlapping subsets (approximately of equal size). For each i $(1 \leq i \leq k)$ the ith subset is kept as validation set while the rest $k-1$ subsets are used for training the model. The process is followed for all the subsets and the average test error rate is estimated. We used `cv.glmnet()` function from the `glmnet` package in R.

2.2.7 Model evaluation : Confusion matrix

		Observed	
		Present	Absent
Predicted	Present	a True positive	b False positive
	Absent	c False negative	d True negative

- Sensitivity is the proportion of true positives correctly identified.

$$\text{Sensitivity} = \frac{a}{a+c}$$

- Precision is the proportion of true positives predicted correctly.

$$\text{Precision} = \frac{a}{a+b}$$

- Specificity is the proportion of true negatives correctly identified.

$$\text{Specificity} = \frac{d}{b+d}$$

Chapter 3

Material and methods

3.1 Data analysis

In this section, we describe the details of the data analysis plan. We critically discuss the choice of different strategies to build the models on presence only data for *Mikania micrantha*.

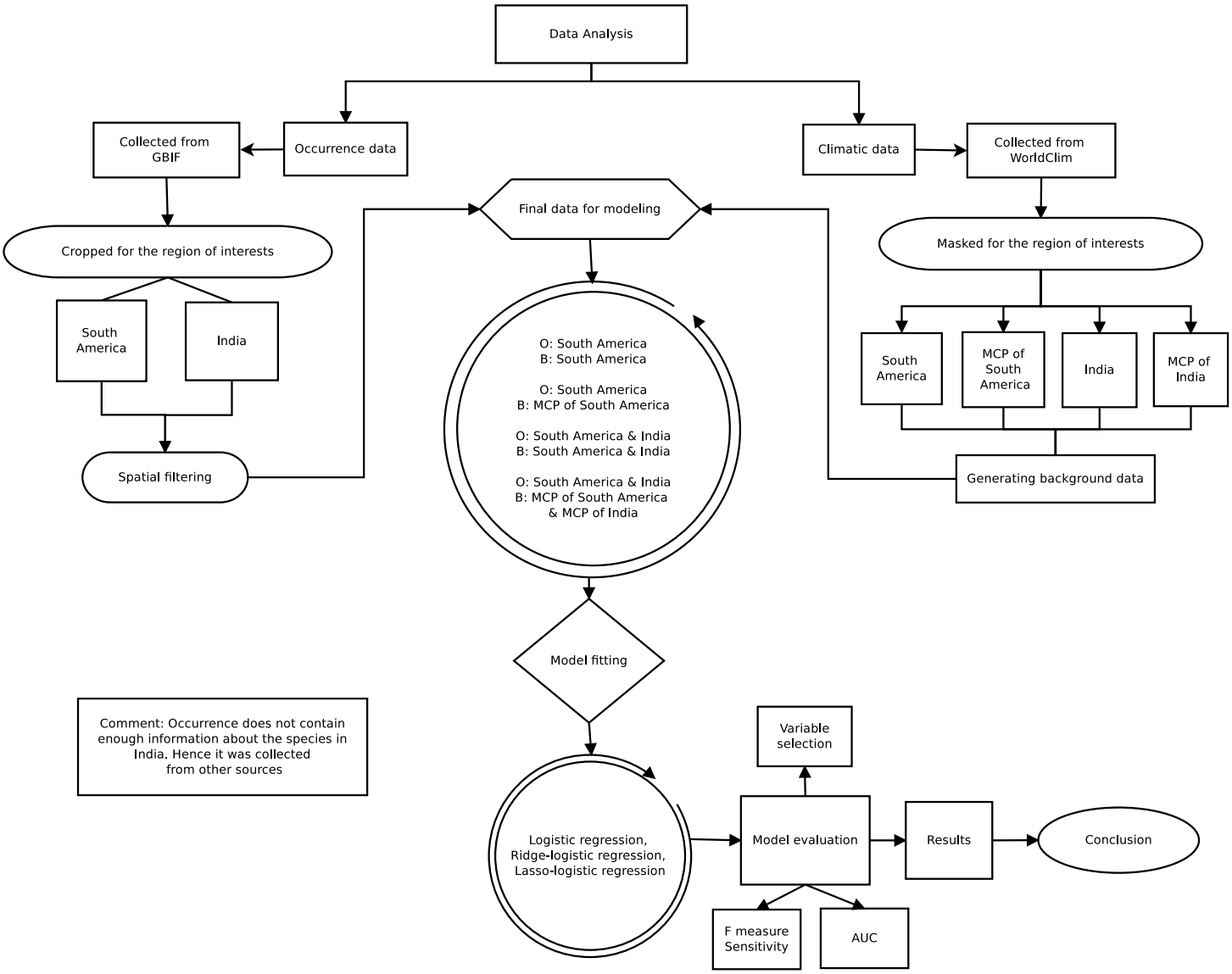


Figure 3.1: Flowchart for data analysis

In this section, we shall describe the complete data analysis plan for the species *"Mikania micrantha Kunth"*.

Occurrence data

The occurrence data on *Mikania micrantha* are collected from different sources which are already demonstrated in Section 2.1. We have accessed the database on January 16, 2017 through the `gbif()` function available in `dismo` package. The query string for genus *"Mikania"* and the species *"micrantha Kunth"* in the `gbif()` function give a total of 3703 records. We have only downloaded the species name, longitude and latitude information. Using the R codes described in Section 2.1, we have treated the missing values and duplicates. *Mikania micrantha* is native to South America, and India is in alien range. We cropped the data using the extent of South America to obtain the native occurrence points. To take care of the geographic bias, we have created fish-net with resolution 2.5. In each grid, only one observation is retained and the others are removed from the data. This process ensures that no particular place(road-side, easily accessible areas etc.) has been preferred by the investigator, which may potentially bias our analysis. Since we are interested to the potential distribution of *Mikania micrantha* in India, we have cropped the data for the extent of India. Unfortunately, GBIF does not contain any information about the species in India. We have obtained the occurrence points in India from Dr. Achyut Kumar Banerjee, Indian Statistical Institute, Kolkata. After the complete data processing, we have 865 occurrence points from South America and 105 from India.

Bioclimatic variables

The 19 environmental covariates are obtained from WorldClim at a resolution of 2.5 arc min. The details of the bioclimatic variables are given in the following:

BIO1 = Annual Mean Temperature

BIO2 = Mean Diurnal Range (Mean of monthly (max temp - min temp))

BIO3 = Isothermality (BIO2/BIO7) (* 100)

BIO4 = Temperature Seasonality (standard deviation *100)

BIO5 = Max Temperature of Warmest Month

BIO6 = Min Temperature of Coldest Month

BIO7 = Temperature Annual Range (BIO5-BIO6)

BIO8 = Mean Temperature of Wettest Quarter

BIO9 = Mean Temperature of Driest Quarter

BIO10 = Mean Temperature of Warmest Quarter

BIO11 = Mean Temperature of Coldest Quarter

BIO12 = Annual Precipitation

BIO13 = Precipitation of Wettest Month

BIO14 = Precipitation of Driest Month

BIO15 = Precipitation Seasonality (Coefficient of Variation)

BIO16 = Precipitation of Wettest Quarter

BIO17 = Precipitation of Driest Quarter

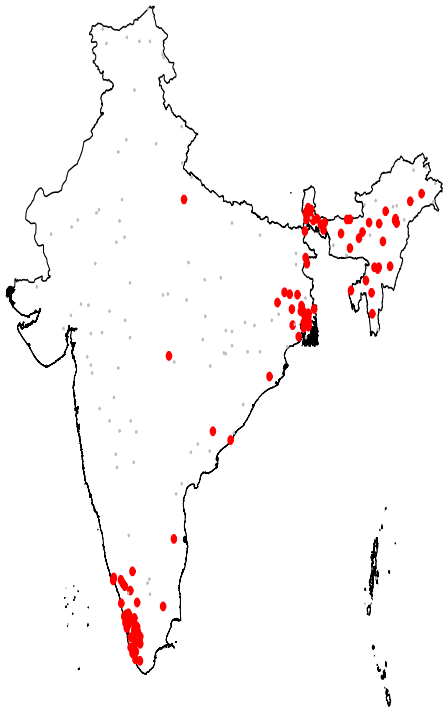
BIO18 = Precipitation of Warmest Quarter

BIO19 = Precipitation of Coldest Quarter

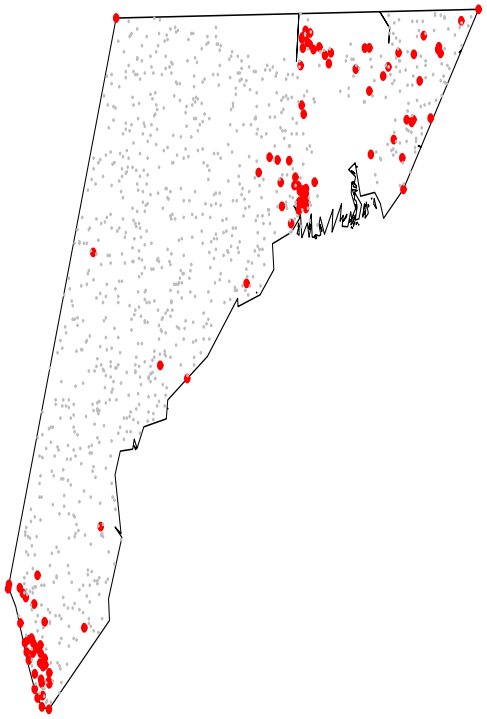
Choice of backgrounds

In our data analysis, we considered four different backgrounds, namely South America, MCP of South America, South America and India & MCP of South America and MCP of India. The environmental covariates are masked accordingly for the regions of interest. For each occurrence points, the values of 19

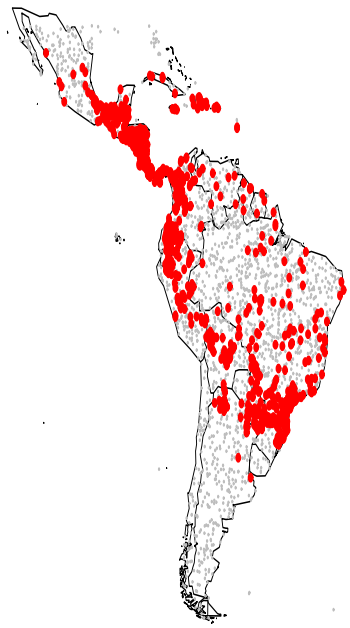
environmental covariates are extracted. For building the species distribution modeling, we generate background points randomly(from each background type) equal to the number of occurrence points. The environmental covariates are also extracted for background points. For example, the final data to build species distribution modeling with South America background will contain 20 columns, 865×2 rows.



(a) India



(b) Minimum convex polygon- India



(c) South America

(d) Minimum convex polygon- South America

Figure 3.2:

Choice of threshold

Sampling large number of random background points(say 10000) with a very low event rate of 8.65 (i.e 865 occurrence points) may not give an accurate result and the threshold must be defined objectively in such case. Hence we select equal number of background points and occurrence points and set the threshold to 0.5.

Model evaluation

To identify the potential distribution of *Mikania micrantha*, we build logistic, ridge-logistic and lasso-logistic regression. The comparative discussion of these methods are already discussed in Section 2.2. To evaluate the model performance, the final data is divided into training set and test set. Since we do not have precise information regarding the presence or absence status at background, during evaluation of the model performance, precision should be under-weighted whereas sensitivity should be given more importance (depicted in Table 2.2.7). Similarly, specificity talks about the proportion of true absences correctly identified in the presence-background data. Due to the lack of precise knowledge about absence points in background sample, specificity should be given less importance. The model performance is evaluated using F measure, which is a combination of sensitivity and precision which is given by,

$$F_\beta = \frac{(1+\beta^2)\cdot \text{precision} \cdot \text{sensitivity}}{\beta^2 \cdot \text{precision} + \text{sensitivity}}.$$

The relative importance of sensitivity and precision is determined with the value of the parameter β. $0 < \beta < 1$ gives more weight to precision while $\beta > 1$ give more weight to the sensitivity. Here we shall use $\beta = 2$.
Using the training set, the models were fitted and the test set were used to test model's predictive accuracy. This process is continued for 10 different randomly chosen background points and average F measure is computed.

The model with highest F measure is declared to be the best model. For each background a total of 300 models were fitted and combining all the backgrounds, the number of fitting is 1200. After fitting the logistic regression model to a particular background sample points we recorded the variables which are significant at 5% level. The process is carried out for each training set and for each background. The same thing is also done for lasso and ridge regression. The frequency of the predictors selected in overall runs are depicted in 4. Since the `glmnet` package does not provide inference about the coefficients, so the coefficient estimates with absolute value less than 10^{-4} are not recorded in the model.

Chapter 4

Results and discussion

The summary of the fits for each model type and different backgrounds are depicted in Figure 4. Similarly for sensitivity measures also, model performance over different backgrounds are depicted in Figure 4.

Plots for F measure

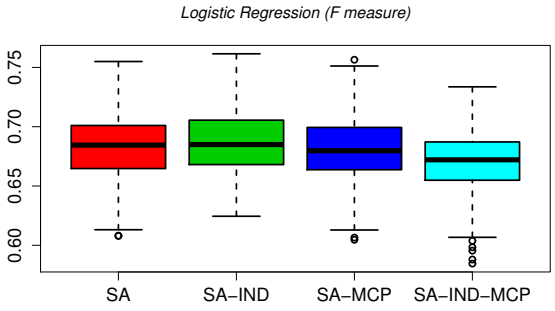


Figure 4.1: F measure for different backgrounds using logistic regression

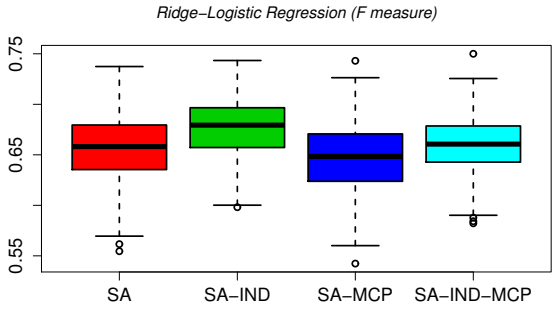


Figure 4.2: F measure for different backgrounds using ridge regression

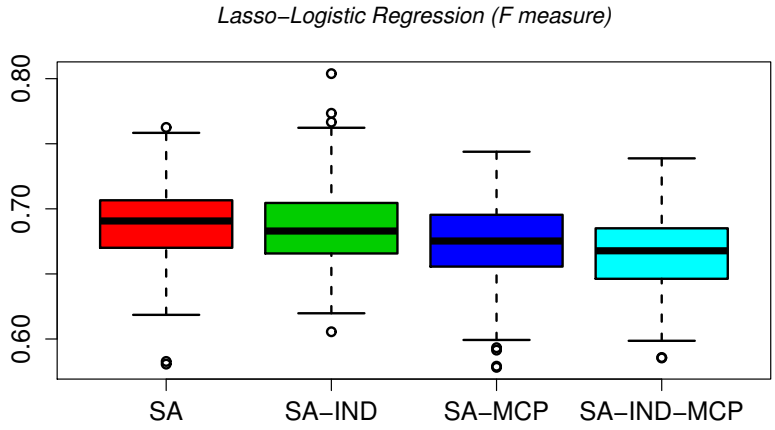


Figure 4.3: F measure for different backgrounds using lasso regression

Plots for Sensitivity

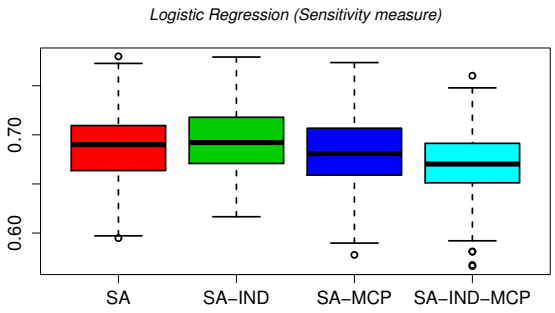


Figure 4.4: Sensitivity for different backgrounds using logistic regression

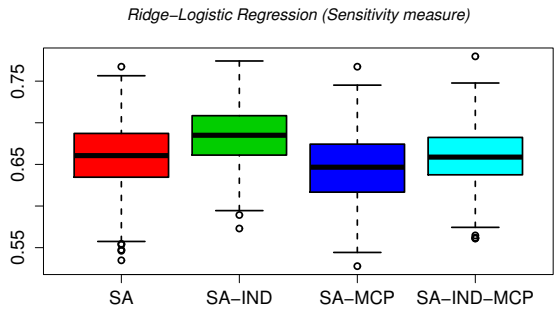


Figure 4.5: Sensitivity for different backgrounds using ridge regression

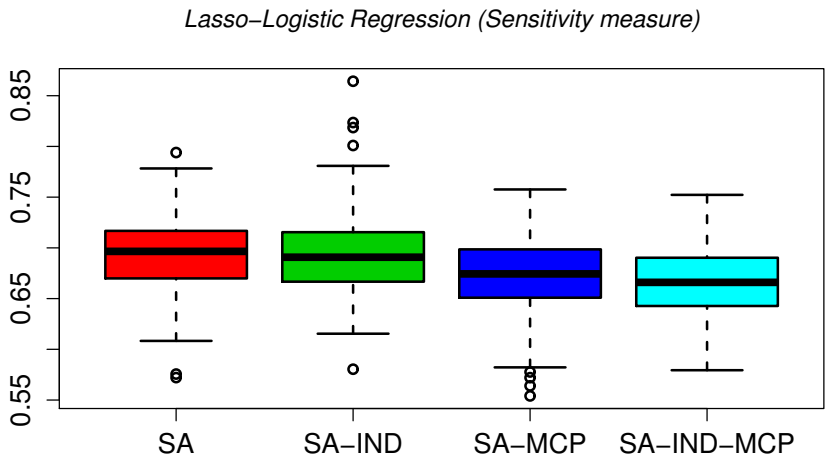
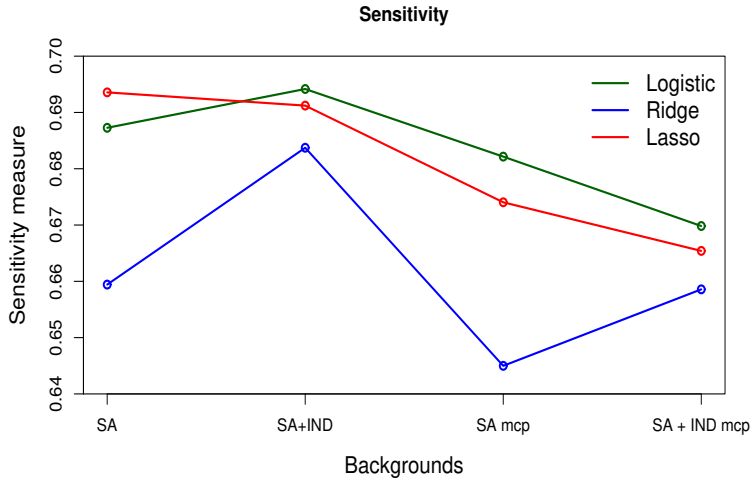
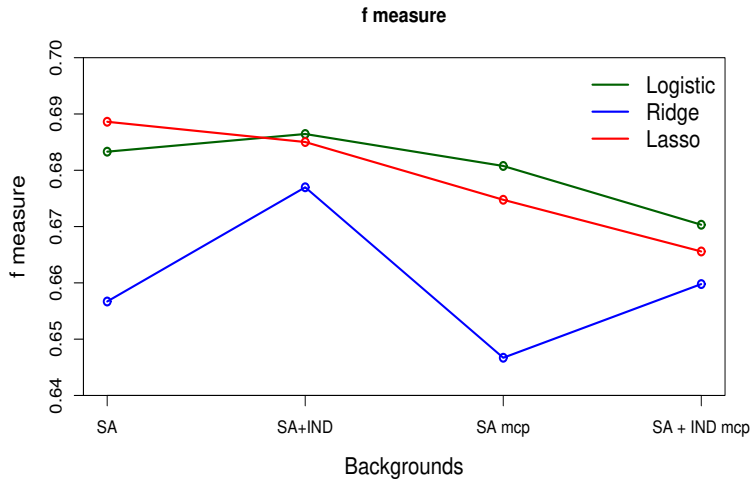


Figure 4.6: Sensitivity for different backgrounds using lasso regression

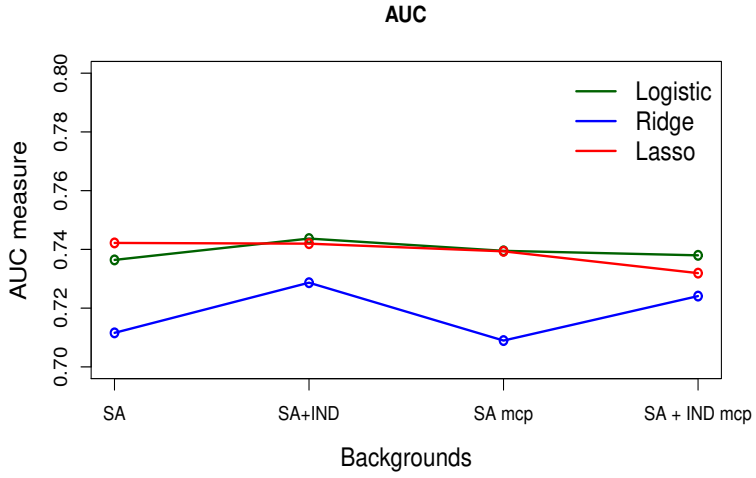
Plot for sensitivity obtained for different backgrounds using Logistic, Ridge and Lasso regression



Plot for F measures obtained for different backgrounds using Logistic, Ridge and Lasso regression



Plot for AUC measure obtained for different backgrounds using Logistic, Ridge and Lasso regression



Backgrounds	Methods	bio1	bio2	bio3	bio4	bio5	bio6	bio7	bio8	bio9	bio10	bio11	bio12	bio13	bio14	bio15	bio16	bio17	bio18	bio19
South America	Logistic	144	300	300	27	300	300		4	46	9	23	215	300	13	170	300	51	267	184
	Ridge	0	300	300	0	255	5	297	268	42	22	0	0	150	52	300	0	0	262	0
	Lasso	4	247	241	0	81	7	215	183	162	86	1	4	220	87	226	199	49	282	12
South America & India	Logistic	46	299	300	22	300	289		14	250	10	0	244	300	7	122	299	28	165	141
	Ridge	38	298	296	0	296	65	297	293	261	0	1	0	144	73	297	0	0	228	0
	Lasso	34	191	200	0	31	0	205	156	148	13	1	2	132	64	199	61	5	227	2
South America MCP	Logistic	106	300	300	35	300	300		23	44	7	10	262	300	6	141	300	69	256	235
	Ridge	15	300	300	0	226	2	159	14	22	1	44	0	37	2	300	0	0	28	0
	Lasso	32	240	295	31	94	65	265	231	154	21	9	21	272	124	289	239	96	296	88
South America MCP & India MCP	Logistic	8	292	300	80	270	300		64	84	58	8	289	300	15	48	300	43	107	222
	Ridge	35	300	299	0	299	22	296	19	225	13	102	0	52	27	300	0	0	31	0
	Lasso	1	230	292	0	254	4	196	26	74	2	15	7	184	28	286	59	16	206	1

Figure 4.7: The above table is built using the number of selected predictors by using the three designated methods for each background. The dark green shades represent strong environmental correlates of species' presence, medium green shades represent the moderate correlates and lighter shades represent correspond to the variables which are rejected more than 50% times in all model runs. The variable BIO7 is not included in the model by logistic regression due to multi collinearity. This is represented by the magenta colored boxes in the above table.

4.1 Model comparison

The performance of the three modeling techniques are evaluated using sensitivity, f measure and AUC. For South America background (native to Mikania), the lasso regression gives highest accuracy as compared to logistic and ridge. In all the models and for all backgrounds, performance of ridge regression is poor as compared to other methods. For complete South America and India background, lasso and logistic have similar accuracy. For the other two backgrounds using minimum convex polygon, logistic performs marginally better than lasso. The advantage of lasso is that along with reasonable accuracy, it is able to select the important environmental covariates described below.

4.2 Selection of environmental covariates

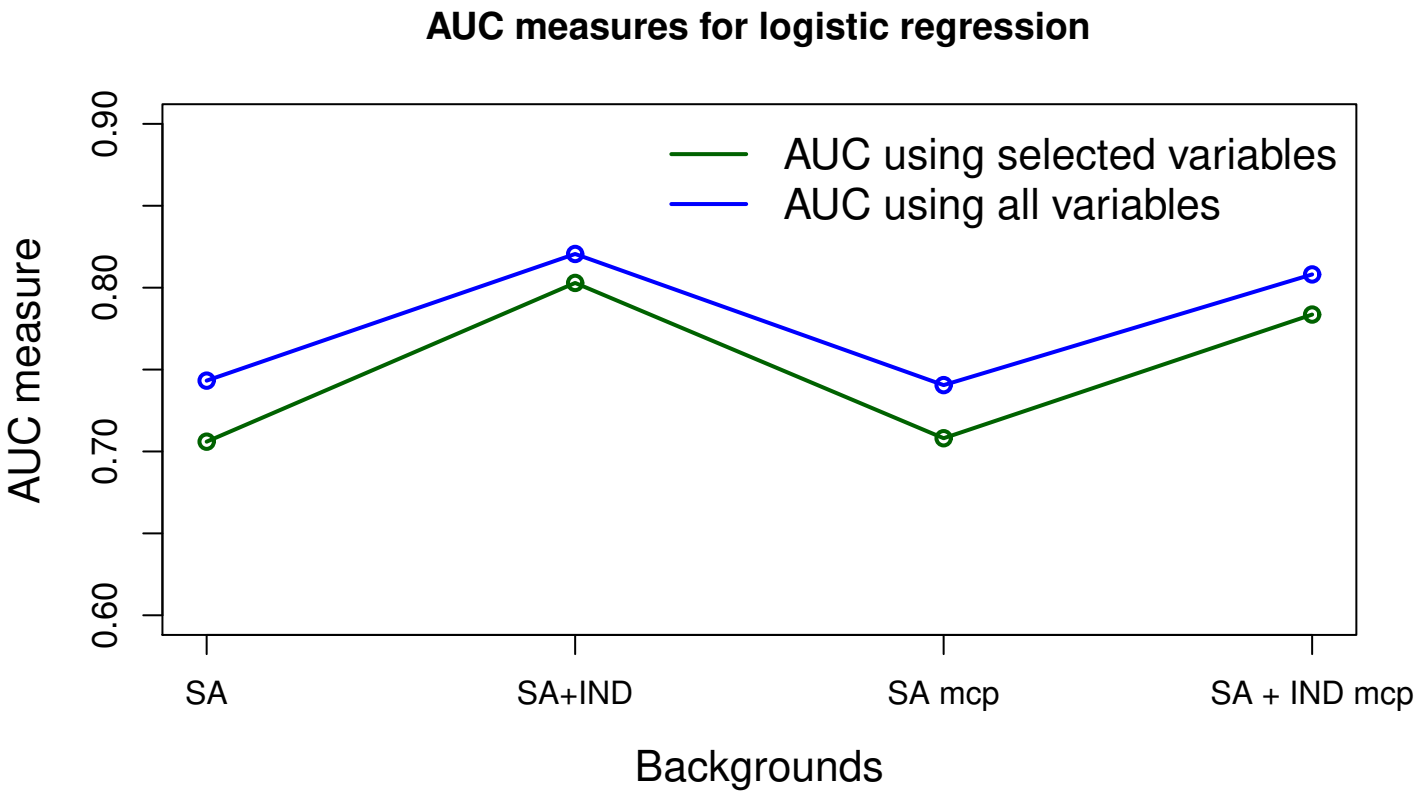
In the earlier sections, we have already discussed the distinction between threshold dependent and threshold independent test. To avoid the problems in choosing threshold we have fixed the number of background points to be selected in each run of the model. The number of background points are kept fixed and equal to the number of occurrences which makes the final data balanced. We have also evaluated the model performance using area under the curve. Since the goal of the project is to identify the environmental factors that better describe the suitability of species occurrence in a particular geographical extent. The existing niche modelling softwares like MaxEnt uses a large number of background points to create the environmental space. Since we are generating only 875 background points, it does not really represent the complete environmental space, part of which is expected to be the niche of Mikania micrantha. To over come this problem, we run the model a large number of times on training set and validate on test set. Since in every run background points are generated randomly the important environmental co-

variates should be statistically significant in most of the runs. We categorize the selected covariates in two sets. The variables which appears statistically significant in more than 80% of the runs are considered to be strong correlates of species realized niche. The variables that appears 50-80% are considered as moderate correlates.

Following the above process and using lasso penalty, mean diurnal range (mean of monthly (max temp - min temp)) (**BIO2**), Isothermality (BIO2/BIO7) (* 100) (**BIO3**), Temperature Annual Range (BIO5-BIO6) (**BIO7**), Precipitation of Wettest Month (**BIO13**), Precipitation Seasonality (Coefficient of Variation) (**BIO15**) and Precipitation of Warmest Quarter (**BIO18**) are found to be strong correlates for all four backgrounds. It is important to note that Max Temperature of Warmest Month (**BIO5**) appeared to be an important environmental factor in minimum convex polygon of South America and minimum convex polygon of India. This may be due to the niche shift of Mikania micrantha from tropical mesic to temperate forest of easter Asia(**?**). Combining the analysis for both native and invaded range, these seven variables are selected. It is worthwhile to mention that **BIO5** is selected as moderate correlate at all backgrounds using ridge penalty.

To verify our findings we build two models using 10000 background points. One model uses all the environmental predictors and the other uses the seven variables selected through lasso. AUC was used to evaluate the models' accuracy.

For both the models, average AUC is computed using 10 training sets with 10000 background points. The relative percentage difference of the two


AUC measures for logistic regression
AUC measure
0.60
0.70
0.80
0.90
AUC using selected variables
AUC using all variables
SA
SA+IND
SA mcp
SA + IND mcp
Backgrounds

methods is measured using

$$\mathrm{d} = 100 \cdot \left(\frac{\mathrm{AUC}_{\mathrm{selected}} - \mathrm{AUC}_{\mathrm{all}}}{\mathrm{AUC}_{\mathrm{all}}} \right)$$

, where $\mathrm{AUC}_{\mathrm{selected}}$ is the AUC measure obtained using the selected variables in the model and $\mathrm{AUC}_{\mathrm{all}}$ is the AUC measure obtained using all of the variables . The average AUC score using selected variables in the model is remarkably close to the full model. For background South America the relative percentage difference is 5.01% only. The percentage difference for other background is depicted in the table below.

	$\mathrm{AUC}_{\mathrm{all}}$	$\mathrm{AUC}_{\mathrm{selected}}$	Relative percentage difference
SA	0.74324	0.70598	5.0137
SA + IND	0.82057	0.802885	2.1587
SA MCP	0.74051	0.70808	4.3787
SA MCP + IND MCP	0.80809	0.78359	3.0324

4.3 Discussion

With the recent advancement in machine learning research, the use of regularization techniques has increased several folds, in particular analyzing high dimensional data where the computation involves sparse matrices. These methods are also becoming equally popular among applied ecologists, in particular, several methodological papers have already appeared in the literature either targeting to improve the general methodological understanding or to better understand the biotic/abiotic interactions of a particular species (ref). However one should be careful in applying these methods that may require the validation of the underlying assumptions for the best application of them. For example, under the true biological scenario if the irrelevant predictors are highly correlated with the variables in the true model then any amount of regularization may not be useful for the selection of true predictors given any

amount of data. The authors described the weak unrepresentable condition as a sufficient criteria that should be empirically checked. The criteria is given by,

$$|C_{2,1} \cdot C_{1,1}^{-1} \cdot \text{sign}(\beta)| < 1$$

where $C_{1,1}$ is the covariance matrix of the important predictors only, $C_{2,1}$ is the covariance between relevant and irrelevant variables.

$$\text{sign}(\beta) = \begin{cases} -1, & \text{if } \beta \text{ is negative} \\ 0, & \text{if } \beta = 0 \\ 1, & \text{if } \beta \text{ is positive} \end{cases}$$

We have checked this condition in every run for the selected variables by lasso regression. We observed that when the selected seven variables were present in the model, the unrepresentable conditions were satisfied empirically.

Existence of nonlinear effect of predictors with the predicted probabilities of occurrence may significantly impact the result. In each cases the model might select irrelevant predictors but more alarmingly may reject an important variable from the model due to its inability to capture the nonlinearity Hastie et al. (2009).

The predictive ability of the models might be improved by employing dimension reduction techniques such as principal component analysis (PCA). The principal components are orthogonal hence the collinearity problem does not appear. But since the variables are clubbed it cannot be used to select the important environmental factors.

4.4 R program for data analysis

```
library(raster)
library(dismo)
library(glmnet)
setwd("G:/My Project/R-Codes/rds files")
bioclim_data_america = brick("bioclim_data_america.grd")
bioclim_data_india = brick("bioclim_data_india.grd")
bioclim_data_india_mcp = brick("bioclim_data_india_mcp.grd")
bioclim_data_mcp = brick("bioclim_data_mcp.grd")
mic_gbif_america  = read.csv("mic_gbif_america.csv", header = TRUE)
predictors_values_america =
read.csv("predictors_values_america.csv", header = TRUE)
predictors_values_america_india =
read.csv("predictors_values_america_india.csv", header = TRUE)
predictors_values_india =
read.csv("predictors_values_india.csv", header = TRUE)
n = 10 # Number of background to be chosen
run = 30 # Number of time the model should run for average test error
test.error = matrix(NA, nrow = run, ncol = n)
bestlm = matrix(NA, nrow = run, ncol = n)
f.measure = matrix(NA, nrow = run, ncol = n)
s = matrix(NA, nrow = run, ncol = n)
background_type = "sa_in"  #sa_mcp, sa_in, sa_in_mcp
# sa = South America;
# sa_mcp = South America minimum convex polygon
# sa_in = South America and India
# sa_in_mcp = South America mcp and India mcp
################################################
```

```
#####   Run of the logistic regression     ######
#################################################
for(i in 1:n)
{
if(background_type == "sa")
 {
  background =
  randomPoints(bioclim_data_america, nrow(predictors_values_america))
  pseudo_absence_values = extract(bioclim_data_america, background)
  y = as.factor(c(rep(1,nrow(predictors_values_america)),
  rep(0,nrow(pseudo_absence_values))))
  sdm_data_america =
  cbind(y,rbind(predictors_values_america,pseudo_absence_values))
 }
if(background_type == "sa_mcp")
 {
  background =
  randomPoints(bioclim_data_mcp, nrow(predictors_values_america))
  pseudo_absence_values = extract(bioclim_data_mcp, background)
  y = as.factor(c(rep(1, nrow(predictors_values_america)),
  rep(0, nrow(pseudo_absence_values))))
  sdm_data_america =
  cbind(y,rbind(predictors_values_america,pseudo_absence_values))
 }
if(background_type == "sa_in_mcp")
 {
  background.america =
  randomPoints(bioclim_data_mcp, nrow(predictors_values_america))
  background.india =
```

```
  randomPoints(bioclim_data_india_mcp,nrow(predictors_values_india))
  pseudo_absence_values.america =
  extract(bioclim_data_mcp, background.america)
  pseudo_absence_values.india =
  extract(bioclim_data_india_mcp, background.india)
  pseudo_absence_values =
  rbind(pseudo_absence_values.america, pseudo_absence_values.india)
  y = as.factor(c(rep(1, nrow(predictors_values_america_india)),
  rep(0, nrow(pseudo_absence_values))))
  sdm_data_america = cbind(y, rbind(predictors_values_america_india,
  pseudo_absence_values))
 }
if(background_type == "sa_in")
 {
  background.america =
  randomPoints(bioclim_data_america, nrow(predictors_values_america))
  background.india =
  randomPoints(bioclim_data_india,nrow(predictors_values_india))
  pseudo_absence_values.america =
  extract(bioclim_data_america, background.america)
  pseudo_absence_values.india =
  extract(bioclim_data_india, background.india)
  pseudo_absence_values = rbind(pseudo_absence_values.america,
  pseudo_absence_values.india)
  y = as.factor(c(rep(1, nrow(predictors_values_america_india)),
  rep(0, nrow(pseudo_absence_values))))
  sdm_data_america =
 cbind(y,rbind(predictors_values_america_india,pseudo_absence_values))
 }
```

```
 for(j in 1:run)
 {
  train = sample(1:nrow(sdm_data_america),
  size = floor(0.75*nrow(sdm_data_america)), replace = FALSE)
  train_data = sdm_data_america[train,]
  test_data = sdm_data_america[-train,]
  test_y = test_data$y
  glm.fit = glm(y ~ ., data = train_data, family = binomial)
  print(summary(glm.fit))
  print(coef(glm.fit))
  glm.probs = predict(glm.fit, test_data, type = "response")
  glm.pred = rep("0", nrow(test_data))
  glm.pred[glm.probs > 0.5] = "1"
  m = as.matrix(table(glm.pred, test_y))
  test.error[j, i] = mean(glm.pred != test_y)
  s[j,i] = m[2,2]/(m[1,2]+m[2,2]); p = m[2,2]/(m[2,1]+m[2,2])
  f.measure[j,i] = 5*p*s[j,i]/(4*p+s[j,i])
 }
}
if(background_type == "sa"){
  write.table(f.measure, "output/fmeasure_logistic_sa_full.txt",
  append = FALSE)
  write.table(s, "output/sensitivity_logistic_sa_full.txt",
  append = FALSE)
}
if(background_type == "sa_in"){
  write.table(f.measure, "output/fmeasure_logistic_sa_in_full.txt",
  append = FALSE)
  write.table(s, "output/sensitivity_logistic_sa_in_full.txt",
```

```
  append = FALSE)
}
if(background_type == "sa_mcp"){
  write.table(f.measure, "output/fmeasure_logistic_sa_mcp.txt",
  append = FALSE)
  write.table(s, "output/sensitivity_logistic_sa_mcp.txt",
  append = FALSE)
}
if(background_type == "sa_in_mcp"){
  write.table(f.measure, "output/fmeasure_logistic_sa_in_mcp.txt",
  append = FALSE)
  write.table(s, "output/sensitivity_logistic_sa_in_mcp.txt",
  append = FALSE)
}
################################################
##### Run of ridge and lasso regression ######
################################################
alpha = 1 # for lasso
#alpha = 0 #for ridge
for(i in 1:n)
{
 if(background_type == "sa")
 {
  background =
  randomPoints(bioclim_data_america, nrow(predictors_values_america))
  pseudo_absence_values = extract(bioclim_data_america, background)
  y = as.factor(c(rep(1, nrow(predictors_values_america)),
  rep(0, nrow(pseudo_absence_values))))
  sdm_data_america =
```

```
 cbind(y, rbind(predictors_values_america, pseudo_absence_values))
}
if(background_type == "sa_mcp")
{
 background =
 randomPoints(bioclim_data_mcp, nrow(predictors_values_america))
 pseudo_absence_values = extract(bioclim_data_mcp, background)
 y = as.factor(c(rep(1, nrow(predictors_values_america)),
 rep(0, nrow(pseudo_absence_values))))
 sdm_data_america =
 cbind(y, rbind(predictors_values_america, pseudo_absence_values))
}
if(background_type == "sa_in_mcp")
{
 background.america =
 randomPoints(bioclim_data_mcp, nrow(predictors_values_america))
 background.india =
 randomPoints(bioclim_data_india_mcp,nrow(predictors_values_india))
 pseudo_absence_values.america =
 extract(bioclim_data_mcp, background.america)
 pseudo_absence_values.india =
 extract(bioclim_data_india_mcp, background.india)
 pseudo_absence_values =
 rbind(pseudo_absence_values.america, pseudo_absence_values.india)
 y = as.factor(c(rep(1, nrow(predictors_values_america_india)),
 rep(0, nrow(pseudo_absence_values))))
 sdm_data_america = cbind(y, rbind(predictors_values_america_india,
 pseudo_absence_values))
}
```

```
if(background_type == "sa_in")
{
 background.america =
 randomPoints(bioclim_data_america, nrow(predictors_values_america))
 background.india =
 randomPoints(bioclim_data_india,nrow(predictors_values_india))
 pseudo_absence_values.america =
 extract(bioclim_data_america, background.america)
 pseudo_absence_values.india =
 extract(bioclim_data_india, background.india)
 pseudo_absence_values =
 rbind(pseudo_absence_values.america, pseudo_absence_values.india)
 y = as.factor(c(rep(1, nrow(predictors_values_america_india)),
 rep(0, nrow(pseudo_absence_values))))
 sdm_data_america =
cbind(y,rbind(predictors_values_america_india,pseudo_absence_values))
}
x = model.matrix(y~., data = sdm_data_america)[,-1]
y = sdm_data_america$y
grid = 10^seq(10, -2, length = 100)
for(j in 1:run)
{
 train = sample(1:nrow(x), size = floor(0.75*nrow(x)),
 replace = FALSE)
 test = (-train)
 test_y = y[test]
 cv.out = cv.glmnet(x = x[train,], y = as.factor(y[train]),
 family = "binomial", alpha=alpha, type.measure = "class")
 bestlm[j,i] = cv.out$lambda.min
```

```
  ridge.probs = predict(cv.out, s=cv.out$lambda.min, newx=x[test,],
  family = "binomial", type = "response")
  ridge.pred = rep(0, length(y[test]))
  ridge.pred[ridge.probs > 0.5] = 1
  m = as.matrix(table(ridge.pred, y[test]))
  test.error[j, i] = mean(ridge.pred != y[test])
  s[j,i] = m[2,2]/(m[1,2]+m[2,2]); p = m[2,2]/(m[2,1]+m[2,2])
  f.measure[j,i] = 5*p*s[j,i]/(4*p+s[j,i])
 }
}
if(alpha == 0)
{
 if(background_type == "sa")
 {
  write.table(f.measure, "output/fmeasure_ridge_sa_full.txt",
  append = FALSE)
  write.table(s, "output/sensitivity_ridge_sa_full.txt",
  append = FALSE)
 }
 if(background_type == "sa_in")
 {
  write.table(f.measure, "output/fmeasure_ridge_sa_in_full.txt",
  append = FALSE)
  write.table(s, "output/sensitivity_ridge_sa_in_full.txt",
  append = FALSE)
 }
 if(background_type == "sa_mcp")
 {
  write.table(f.measure, "output/fmeasure_ridge_sa_mcp.txt",
```

```
  append = FALSE)
  write.table(s, "output/sensitivity_ridge_sa_mcp.txt",
  append = FALSE)
 }
 if(background_type == "sa_in_mcp")
 {
  write.table(f.measure, "output/fmeasure_ridge_sa_in_mcp.txt",
  append = FALSE)
  write.table(s, "output/sensitivity_ridge_sa_in_mcp.txt",
  append = FALSE)
 }
}
if(alpha == 1)
{
 if(background_type == "sa")
 {
  write.table(f.measure, "output/fmeasure_lasso_sa_full.txt",
  append = FALSE)
  write.table(s, "output/sensitivity_lasso_sa_full.txt",
  append = FALSE)
 }
 if(background_type == "sa_in")
 {
  write.table(f.measure, "output/fmeasure_lasso_sa_in_full.txt",
  append = FALSE)
  write.table(s, "output/sensitivity_lasso_sa_in_full.txt",
  append = FALSE)
 }
 if(background_type == "sa_mcp")
```

```
 {
  write.table(f.measure, "output/fmeasure_lasso_sa_mcp.txt",
  append = FALSE)
  write.table(s, "output/sensitivity_lasso_sa_mcp.txt",
  append = FALSE)
 }
 if(background_type == "sa_in_mcp")
 {
  write.table(f.measure, "output/fmeasure_lasso_sa_in_mcp.txt",
  append = FALSE)
  write.table(s, "output/sensitivity_lasso_sa_in_mcp.txt",
  append = FALSE)
 }
}
```

Chapter 5

Conclusion and future direction

In this project, we have investigated the use of two regularization techniques(lasso and ridge) in building the species distribution model. We observed that the accuracy of logistic model is more as compared to the lasso and ridge. For combined data from South America and India gave similar accuracy for logistic and lasso. The maximum average F measure is 0.70 and only 69% of true presences are identified correctly. The used techniques in this project do not consider the nonlinear relationships of the predictor variables with the predicted probabilities of occurrences. To capture such hidden nonlinearities more flexible modeling frameworks e.g. generalized additive models, neural network may be more appropriate choice. It would be our future endeavor to explore the utility of these techniques to identify the potential distribution of *Mikania micrantha*.

Chapter 6

Bibliography and references

Bibliography

A. Agresti. *An introduction to categorical data analysis*. Wiley-Blackwell, 2007.

Michael H. Graham. Confronting multicollinearity in ecological multiple regression. *Ecology*, 84(11):2809–2815, 2003. ISSN 1939-9170. doi: 10.1890/02-3114. URL http://dx.doi.org/10.1890/02-3114.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition, 2009. URL http://www-stat.stanford.edu/~tibs/ElemStatLearn/.

William W. Hines, Douglas C. Montgomery, David M. Goldsman, and Connie M. Borror. *Probability and Statistics in Engineering*. John Wiley & Sons, Inc., 4th edition, 2003.

A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 1970a.

A. E. Hoerl and R. W. Kennard. Ridge regression: Applications to nonorthogonal problems. *Technometrics*, 12:69–82, 1970b.

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated, 2014. ISBN 1461471370, 9781461471370.

Farber Oren Kadmon, Ronen and Avinoam Danin. Effect of roadside bias on the accuracy of predictive maps produced by bioclimatic models. *Ecological*

Applications, 14(2):401–413, 2004. ISSN 1939-5582. doi: 10.1890/02-5364. URL http://dx.doi.org/10.1890/02-5364.

Keith Knight and Wenjiang Fu. Asymptotics for lasso-type estimators. *The Annals of Statistics*, 28(5):1356–1378, 2000.

S. le Cessie and J.C. van Houwelingen. Ridge estimators in logistic regression. *Applied Statistics*, 41(1):191–201, 1992.

Douglas C. Montgomery, Elizabeth A. Peck, and Geoffrey G. Vining. *Introduction to Linear Regression Analysis (4th ed.)*. Wiley & Sons, 2006. ISBN 0471754951.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2016. URL https://www.R-project.org/.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B)*, 58:267–288, 1996.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2003.

Conventions used in R codes

The following conventions are used in the R codes of this project:

`mic_gbif_america` consists of longitude-latitude information about the presence locations of the species in the region of South America.
`mic_gbif_india` consists of longitude-latitude information about the presence locations of the species in the Indian subcontinent.
`bioclim_data_world` is a raster file consisting of information about the environmental covariates for the whole world.
`bioclim_data_america` is a raster file consisting of information about the environmental covariates for the region of South America.
`bioclim_data_india` is a raster file consisting of information about the environmental covariates for the region of India.
`bioclim_data_mcp` is a raster file consisting of information about the environmental covariates for the minimum convex polygon built in the region of South America.
`bioclim_data_india_mcp` is a raster file consisting of information about the environmental covariates for the minimum convex polygon built in the region of India.
`predictors_values_america` consists of information about the environmental covariates for the presence locations of species in the region of South America.
`predictors_values_india` consists of information about the environmental covariates for the presence locations of species in the region of India.
`predictors_values_america_india` consists of information about the environmental covariates for the presence locations of species in the region of South America and India.
`wrld_simpl` is a `SpatialPolygonsDataFrame` containing world map.
`pseudo_absence_values` consists of environmental covariates for the back-

ground points generated from the geographical extent that we have considered.

`sdm_data_america` is the final data that we use for modeling. It consists of a response variable with binary outcome(either 0 or 1) and the 19 explanatory variables(environmental covariates) of presence and background locations.

Additional R codes for figures

```
####################################
####### LASSO REGRESSION ###########
####################################
> fmeasure_lasso_sa_full =
        read.table("fmeasure_lasso_sa_full.txt")
> fmeasure_lasso_sa_in_full =
        read.table("fmeasure_lasso_sa_in_full.txt")
 fmeasure_lasso_sa_mcp =
        read.table("fmeasure_lasso_sa_mcp.txt")
 fmeasure_lasso_sa_in_mcp =
        read.table("fmeasure_lasso_sa_in_mcp.txt")
 sensitivity_lasso_sa_full =
        read.table("sensitivity_lasso_sa_full.txt")
 sensitivity_lasso_sa_in_full =
        read.table("sensitivity_lasso_sa_in_full.txt")
 sensitivity_lasso_sa_mcp =
        read.table("sensitivity_lasso_sa_mcp.txt")
> sensitivity_lasso_sa_in_mcp =
        read.table("sensitivity_lasso_sa_in_mcp.txt")
####################################
####### RIDGE REGRESSION ###########
####################################
> fmeasure_ridge_sa_full =
        read.table("fmeasure_ridge_sa_full.txt")
 fmeasure_ridge_sa_in_full =
        read.table("fmeasure_ridge_sa_in_full.txt")
 fmeasure_ridge_sa_mcp =
```

```
        read.table("fmeasure_ridge_sa_mcp.txt")
 fmeasure_ridge_sa_in_mcp =
        read.table("fmeasure_ridge_sa_in_mcp.txt")
 sensitivity_ridge_sa_full =
        read.table("sensitivity_ridge_sa_full.txt")
 sensitivity_ridge_sa_in_full =
        read.table("sensitivity_ridge_sa_in_full.txt")
 sensitivity_ridge_sa_mcp =
        read.table("sensitivity_ridge_sa_mcp.txt")
 sensitivity_ridge_sa_in_mcp =
        read.table("sensitivity_ridge_sa_in_mcp.txt")
##################################
###### LOGISTIC REGRESSION ######
##################################
> fmeasure_logistic_sa_full =
        read.table("fmeasure_logistic_sa_full.txt")
> fmeasure_logistic_sa_in_full =
        read.table("fmeasure_logistic_sa_in_full.txt")
> fmeasure_logistic_sa_mcp =
        read.table("fmeasure_logistic_sa_mcp.txt")
> fmeasure_logistic_sa_in_mcp =
        read.table("fmeasure_logistic_sa_in_mcp.txt")
> sensitivity_logistic_sa_full =
        read.table("sensitivity_logistic_sa_full.txt")
> sensitivity_logistic_sa_in_full =
        read.table("sensitivity_logistic_sa_in_full.txt")
> sensitivity_logistic_sa_mcp =
        read.table("sensitivity_logistic_sa_mcp.txt")
> sensitivity_logistic_sa_in_mcp =
```

```
        read.table("sensitivity_logistic_sa_in_mcp.txt")
#####################################################################
######### BOX-PLOTS FOR F MEASURE AND SENSITIVITY ###################
#####################################################################
> boxplot(as.numeric(unlist(fmeasure_lasso_sa_full)),
        as.numeric(unlist(fmeasure_lasso_sa_in_full)),
        as.numeric(unlist(fmeasure_lasso_sa_mcp)),
        as.numeric(unlist(fmeasure_lasso_sa_in_mcp)),
        col = 10:14, names = c("SA", "SA-IND", "SA-MCP",
    "SA-IND-MCP"), main = list("Lasso-Logistic Regression(F measure)",
    font = 3), lwd =2, cex.axis = 1.3)
> boxplot(as.numeric(unlist(sensitivity_lasso_sa_full)),
        as.numeric(unlist(sensitivity_lasso_sa_in_full)),
        as.numeric(unlist(sensitivity_lasso_sa_mcp)),
        as.numeric(unlist(sensitivity_lasso_sa_in_mcp)), col = 10:14,
        names = c("SA", "SA-IND", "SA-MCP", "SA-IND-MCP"),
        main = list("Lasso-Logistic Regression(Sensitivity measure)",
    font = 3), lwd =2, cex.axis = 1.3)
> logistic_sensitivity =
        c(mean(colMeans(sensitivity_logistic_sa_full)),
        mean(colMeans(sensitivity_logistic_sa_in_full)),
    mean(colMeans(sensitivity_logistic_sa_mcp)),
        mean(colMeans(sensitivity_logistic_sa_in_mcp)))
> ridge_sensitivity =
        c(mean(colMeans(sensitivity_ridge_sa_full)),
        mean(colMeans(sensitivity_ridge_sa_in_full)),
    mean(colMeans(sensitivity_ridge_sa_mcp)),
        mean(colMeans(sensitivity_ridge_sa_in_mcp)))
> lasso_sensitivity =
```

```
        c(mean(colMeans(sensitivity_lasso_sa_full)),
        mean(colMeans(sensitivity_lasso_sa_in_full)),
    mean(colMeans(sensitivity_lasso_sa_mcp)),
        mean(colMeans(sensitivity_lasso_sa_in_mcp)))
> plot(logistic_sensitivity,type = "o",col = "dark green",
        lwd = 2, ylim= c(0.64,0.7), xaxt = "n", xlab = "Backgrounds",
    ylab = "Sensitivity measure", main = "Sensitivity",
        cex.lab = 1.3, cex.axis = 1.1)
> axis(1, at = 1:4, labels = c("SA", "SA+IND", "SA mcp",
        "SA + IND mcp"), cex.lab = 1.3)
> lines(ridge_sensitivity, type = "o",lwd = 2, col = "blue")
> lines(lasso_sensitivity, type = "o", lwd = 2,col = "red")
> legend("topright", col = c("dark green", "blue", "red"),
        cex = 1.3, legend = c("Logistic", "Ridge", "Lasso"),
    bty = "n", lwd = 2)
> logistic_fmeasure =
        c(mean(colMeans(fmeasure_logistic_sa_full)),
        mean(colMeans(fmeasure_logistic_sa_in_full)),
    mean(colMeans(fmeasure_logistic_sa_mcp)),
        mean(colMeans(fmeasure_logistic_sa_in_mcp)))
> ridge_fmeasure =
        c(mean(colMeans(fmeasure_ridge_sa_full)),
        mean(colMeans(fmeasure_ridge_sa_in_full)),
    mean(colMeans(fmeasure_ridge_sa_mcp)),
    mean(colMeans(fmeasure_ridge_sa_in_mcp)))
> lasso_fmeasure =
        c(mean(colMeans(fmeasure_lasso_sa_full)),
        mean(colMeans(fmeasure_lasso_sa_in_full)),
        mean(colMeans(fmeasure_lasso_sa_mcp)),
```

```
        mean(colMeans(fmeasure_lasso_sa_in_mcp)))
> plot(logistic_fmeasure,type = "o",col = "dark green",
        lwd = 2, ylim= c(0.64,0.7), xaxt = "n",
    xlab = "Backgrounds", ylab = "f measure",
    main = "f measure", cex.lab = 1.3,
    cex.axis = 1.1)
> axis(1, at = 1:4, labels = c("SA", "SA+IND", "SA mcp",
        "SA + IND mcp"), cex.lab = 1.3)
> lines(ridge_fmeasure, type = "o",lwd = 2, col = "blue")
> lines(lasso_fmeasure, type = "o", lwd = 2,col = "red")
> legend("topright", col = c("dark green", "blue", "red"),
        cex = 1.3, legend = c("Logistic", "Ridge", "Lasso"),
    bty = "n", lwd = 2)
```