Benton Tripp

GIS 540 Final Project Proposal

1. **Title:** North Carolina Woodpecker Species Distribution Modeling using Presence-Only Prediction
2. **Abstract:** A major challenge in understanding the distribution of species is that observational point data tends to be primarily recorded in locations easily accessible to people. One method of estimating the distribution in a broader area is through a spatial statistics technique called Presence Only Prediction using Maximum Entropy (MaxEnt)1. The purpose of this project is to use the MaxEnt approach to estimate woodpecker species’ distributions in North Carolina from crowd-sourced bird watching dataset.
3. **Input data:** The data used in this project are from three primary sources:
   1. Project FeederWatch2 – Bird observation data, compiled by The Cornell Lab of Ornithology and Birds Canada. There are two file formats for the data: (1) *.xlsx* and (2) *.csv*. The Excel file contains the “data dictionary”, and is primarily needed so that the common names of the woodpecker species can be joined with the *.csv* data (in which they are instead identified by a unique ID). The *.csv* data contains the relevant data needed for this project including the species, date observed, and the location observed. The data can be obtained directly from the Project FeederWatch website via the dataset requests page: <https://feederwatch.org/explore/raw-dataset-requests/>.
   2. National Land Cover Database3 – using data from the Multi-Resolution Land Characteristics (MRLC) Consortium4, North Carolina State University created the NLCD. The data used in this project is clipped/re-projected land cover data for the state of North Carolina. <https://gisdata.lib.ncsu.edu/fedgov/mrlc/nlcd2019/NC_NLCD2019only.zip>
   3. North Carolina Digital Elevation Model (DEM)5 – this data was acquired directly through the NCSU GIS library at: <https://gisdata.lib.ncsu.edu/DEM/nc250.zip>. The data is a raster image of North Carolina elevation at a 1:250,000 scale.

The observation data will serve as the input point features for the MaxEnt models (each species will be modeled separately), while the two raster datasets will be the explanatory rasters.

1. **Data Products:** The original bird observation data is sparse, and tends to be most prevalent in human-accessible areas where birdwatching is popular. The Presence-Only Prediction (MaxEnt) Algorithm will use explanatory variables (i.e., elevation and land cover) to estimate how likely it might be to observe the same birds across the entire study area. The function available through *arcpy* provides several outputs:

* Trained raster (*see Figure 1 of the example from the arcpy Presence-Only Prediction documentation*6)
* Trained features (background point data generated to aid in training the model against the explanatory features)
* Descriptive tables (response curve, sensitivity)

The primary output of concern in the scope of this project is the trained raster, since it can be used to easily visualize the estimated distribution of the input bird species across the study area.

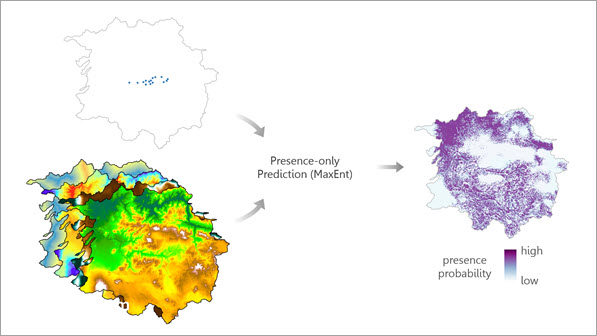
[](https://pro.arcgis.com/en/pro-app/latest/tool-reference/spatial-statistics/presence-only-prediction.htm)

Figure 1: Point data, explanatory raster(s), and other explanatory variables are used as inputs to the Presence-Only Prediction (MaxEnt) algorithm, and a presence probability raster is output for the defined study area. This image was taken from the arcpy Presence-Only Prediction documentation6

1. **Batch processing:** *A description of the processing problem that you want to handle with Python (required).*

The processing problem will include:

1. Pulling in the FeederWatch data (since the data is spread across multiple .csv files by date, this will be done as a batch process). Each individual part of the data will be preprocessed before all of them are combined into a final dataset and saved to a .csv file.
   1. The preprocessing step includes:
      1. Filtering the data to only consist of bird species in the Woodpecker family
      2. Removing possible “bad” records (there is a field that specifies if the record was verified)
      3. Fixing field names
      4. Filter by US State
      5. Filtering by date range
2. Set up environment (i.e., create geodatabase, define projected coordinate system, etc.)
3. For each species in the final data, create a Feature Class and add the defined projected coordinate system to the feature class.
4. Pull in Raster Datasets, and save to the Geodatabase
5. Resample the Land Cover raster to be 1000 x 1000 meters
6. For each species of woodpecker, use the Presence-Only-Prediction tool to estimate the distribution of that species in the state of North Carolina
7. **Pseudocode:** *Pseudocode for the problem described in the previous step (required).*

FUNC get\_species\_codes(species codes url, bird family)

GET species codes from species codes url

COMPUTE filtered species codes by filtering the data

by bird family

RETURN filtered species codes

END FUNC

FUNC clean\_data(raw feeder watch data, filtered species, US

state, minimum year, year)

SET list of all of the correct names

FOR name in raw feeder watch data field names

IF name not in correct names

Rename to the correct name

END IF

END FOR

SET filtered feeder watch data as the raw feeder

watch data filtered by the specified species

SET filtered feeder watch data as the filtered feeder

watch data filtered again, but the specified US

state

SET filtered feeder watch data as the filtered feeder

watch data filtered again, but as greater than or

equal to the minimum year, less than or equal to

the maximum year

SET final feeder watch data as the filtered feeder

watch data where the valid field is set to True

RETURN final feeder watch data

END FUNC

FUNC get\_feeder\_watch\_data(outfile, time frame list,

filtered species, US state,

minimum year, maximum year,

output directory)

SET final output .csv file name

IF final output file exists THEN

READ feeder watch data from output file

ELSE

SET data list

FOR time frame in time frame list

SET intermediate output .csv file name

IF intermediate file does not exist

SET feeder watch data url for the time

frame

GET raw feeder watch data from url

cleaned data = clean\_data(raw feeder

watch data,

filtered

species, US state, minimum year, maximum year)

Save cleaned data to intermediate file

ELSE

READ intermediate file

ENDIF

Append cleaned data to data list

ENDFOR

SET feeder watch data as the combined data list

ENDIF

RETURN feeder watch data

END FUNC

FUNC batch\_db\_import(feeder watch .csv file name, feature

class name, existing feature classes,

projected coordinate system, output

directory, feeder watch data, filtered

species data)

IF base feature class name not in existing feature

classes THEN

Create a new feature class from the feeder watch

.csv file, located at the output directory

ENDIF

IF projected base feature class not in existing

feature classes THEN

Create a new feature class using the base feature class and the projected coordinate system

ENDIF

GET unique species names from feeder watch data

SET species feature class list as empty list

FOR species name in unique species names

SET species feature class name

IF species feature class name not in existing

feature classes THEN

Select the subset of the base feature class (with the projected coordinate system) containing the species name, and create a new feature class called the species feature class name

ENDIF

Append species feature class name to species

feature class name list

ENDFOR

RETURN species feature class list

END FUNC

FUNC get\_rasters(land cover raster url, elevation raster

url, land cover output path, elevation

raster output path, land cover raster

name, elevation raster name, land cover

resample size)

IF land cover output path does not exist THEN

GET land cover raster from land cover raster url

Save to land cover output path

Resample land cover data using land cover

resample size, save to geodatabase as land cover

raster name

ENDIF

IF elevation raster output path does not exist THEN

GET elevation raster from elevation raster url

Save to elevation raster output path

Copy elevation raster to geodatabase as elevation

raster name

ENDFUNC

SET feeder watch .csv file name

SET species codes url

SET bird family

filtered species data = get\_species\_codes(species codes

url, bird family)

SET time frame list

SET minimum year

SET maximum year

SET ouput directory

SET filename suffix

feeder watch data = get\_feeder\_watch\_data(feeder watch .csv

file name, time

frame list,

filtered species

data, US state,

minimum year,

maximum year,

ouput directory,

filename suffix)

SET project path

SET database name

IF database name not in project path THEN

Create the database

ENDIF

SET workspace as database path

GET existing feature classes (might be empty list)

SET projected coordinate system

SET base feature class name

species feature class list = batch\_db\_import(feeder watch

.csv file

name, base

feature class name, existing feature classes, projected coordinate system,

output

directory, feeder watch data, filtered species data)

SET land cover raster url

SET land cover output path

SET land cover raster name

SET land cover resample size

SET elevation raster url

SET elevation raster output path

SET elevation raster name

get\_rasters(land cover raster url, elevation raster url,

land cover output path, elevation raster output

path, land cover raster name, elevation raster

name, and cover resample size)

FOR species feature class in species feature class list

SET MaxEnt output file names

COMPUTE MaxEnt using the species feature class,

elevation raster, and resampled land cover

raster; ouputs are saved to the defined MaxEnt

output file names

ENDFOR

1. **Keywords:** species distribution, raster, point data, digital elevation model, land cover, presence-only prediction, MaxEnt, statistics, birds
2. **Informal context** *(only include this in the first draft of the proposal for internal use):   In this section please share with us the kind of information that you wouldn’t put in a formal project proposal.  Here are some questions that aren’t always made clear but would help us to understand the context in which your project arises*.
3. *My project proposal is related to* (*Three apply, options i-j)*:

h. One of my hobbies.

i. A topic that I found interesting

j. A dataset that I found so that I could have a project topic

1. *Which of the following statements best describes your progress thus far?*

e. I have completed (c) and written more than 50 lines of code. Note that option c. states that I have inspected the data and written my proposal.

1. *Have you worked on this or a similar project for a different class or for work, that you’d like to automate?  If so, which class, and in what format (Modelbuilder? Hand running tools, etc.) Briefly explain.*

I have not worked specifically with ArcGIS doing something like this before, but I use Python and R for my job on a daily basis. I am very familiar with what are referred to in this class as “batch processes”, although my work is typically more in the realm of machine learning and data science.

1. I am currently still uncertain about the following aspects of my project:
   1. ~~The type of the data.~~
   2. ~~The source of the data.~~
   3. ~~The geoprocessing steps that will be needed.~~
   4. ~~The workflow that will be needed.~~
   5. ~~The questions I want to answer.~~
   6. ~~The output I want to generate.~~
   7. Other, please explain. *I am not too familiar with the MaxEnt algorithm and would like to do more research on its functionality so that I can tune the parameters of each model accordingly.*
2. Which of the following statements best describes your feelings toward this project:
   1. I feel very passionate about the project topic I have selected and will commit to coming up with substantial extensions if it is not sophisticated enough in its current form.
3. Any additional informal but pertinent remarks/questions that don’t fit in elsewhere: N/A

**References**

1. <https://pro.arcgis.com/en/pro-app/latest/tool-reference/spatial-statistics/presence-only-prediction.htm>
2. <https://feederwatch.org/about/project-overview/>
3. <https://www.lib.ncsu.edu/gis/nlcd>
4. <https://www.mrlc.gov/>
5. <https://www.lib.ncsu.edu/gis/dem>
6. <https://pro.arcgis.com/en/pro-app/latest/tool-reference/spatial-statistics/presence-only-prediction.htm>