Improving Coarsely Populated Geolocation Data with Inverse Reinforcement Learning

CSCI470: Machine Learning

Seth Asadi, Zach Mills, Alex Pollock, Dominic Quintana, & Leah Reeder

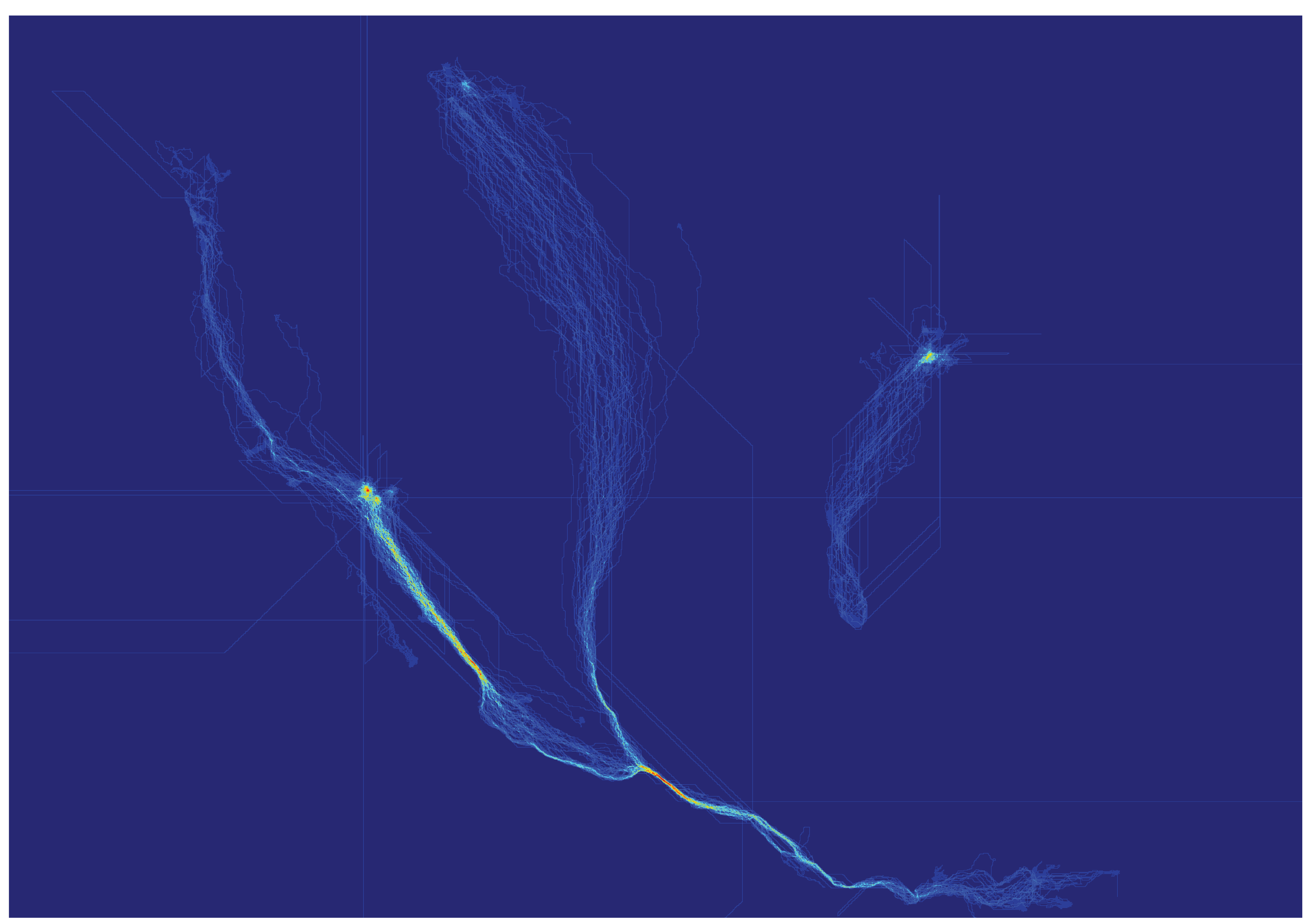
Problem Statement

Ethologists/Ecologists face spatial & temporal bias in behavioral analysis due to coarsely populated location data. These location gaps are due to the obstacles in battery consumption, size, & signal strength that animal-bourne GPS devices face. The current approach of using linear path-interpolation is insufficient because it over-simplifies the path & indirect nature of animal movement.

Data

Figure 1) *Visual of Turkey Vulture migration geolocation data and input training/testing data: Turkey Vulture geolocation data from movebank.org*

Figure 2) *Heatmap visual of Turkey Vulture migration geolocation data.*

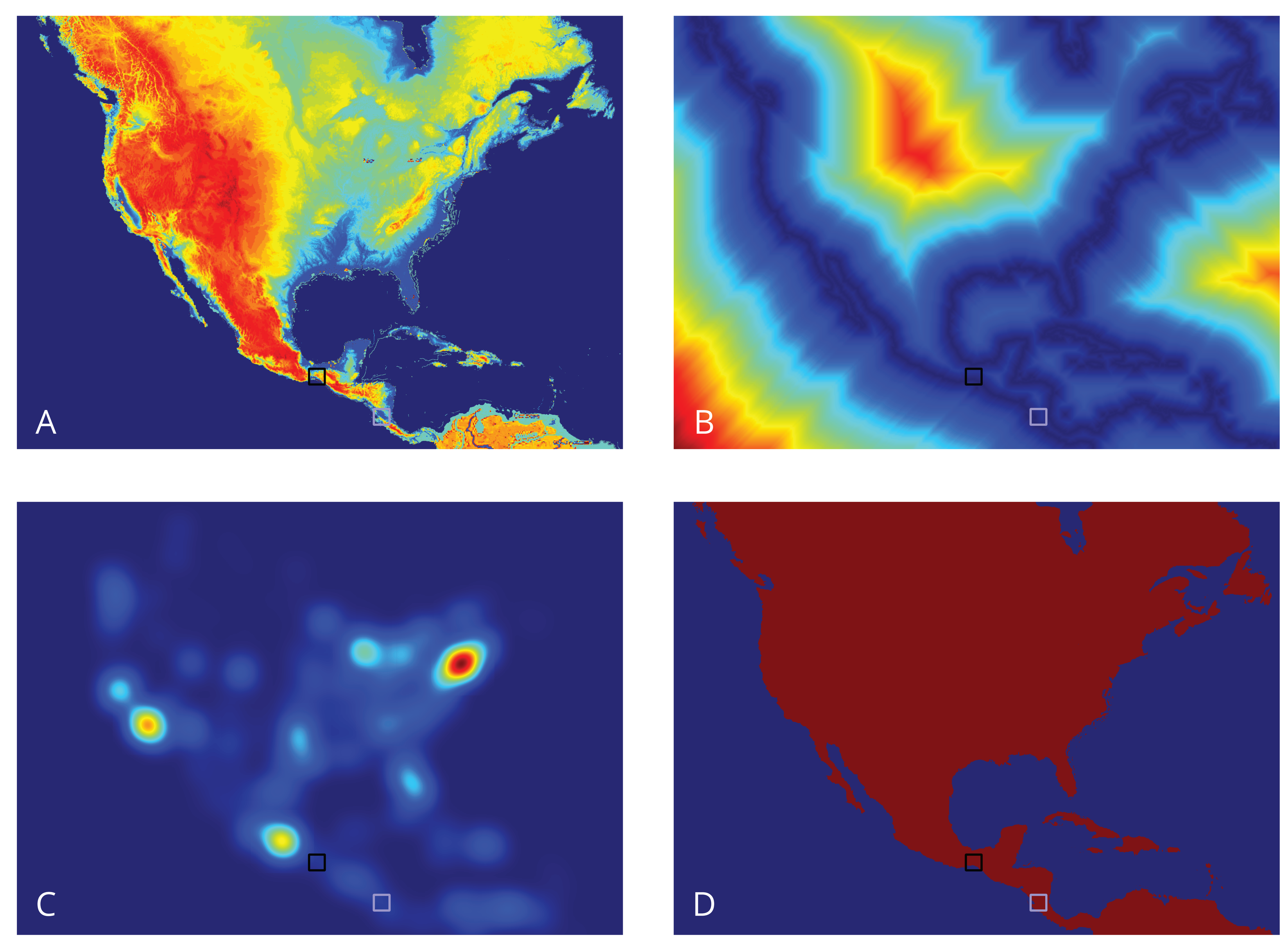


Features

Datasets Used

- elevation
- distance from the coast
- city populations & locations (estimation for population density)
- ocean vs. land

Figure 3) *Generated reward maps of collected features: (A) elevation (B) distance from coast (C) population density and (D) ocean or water. Red indicates a maximum value, blue a minimum.*



Preliminary Results

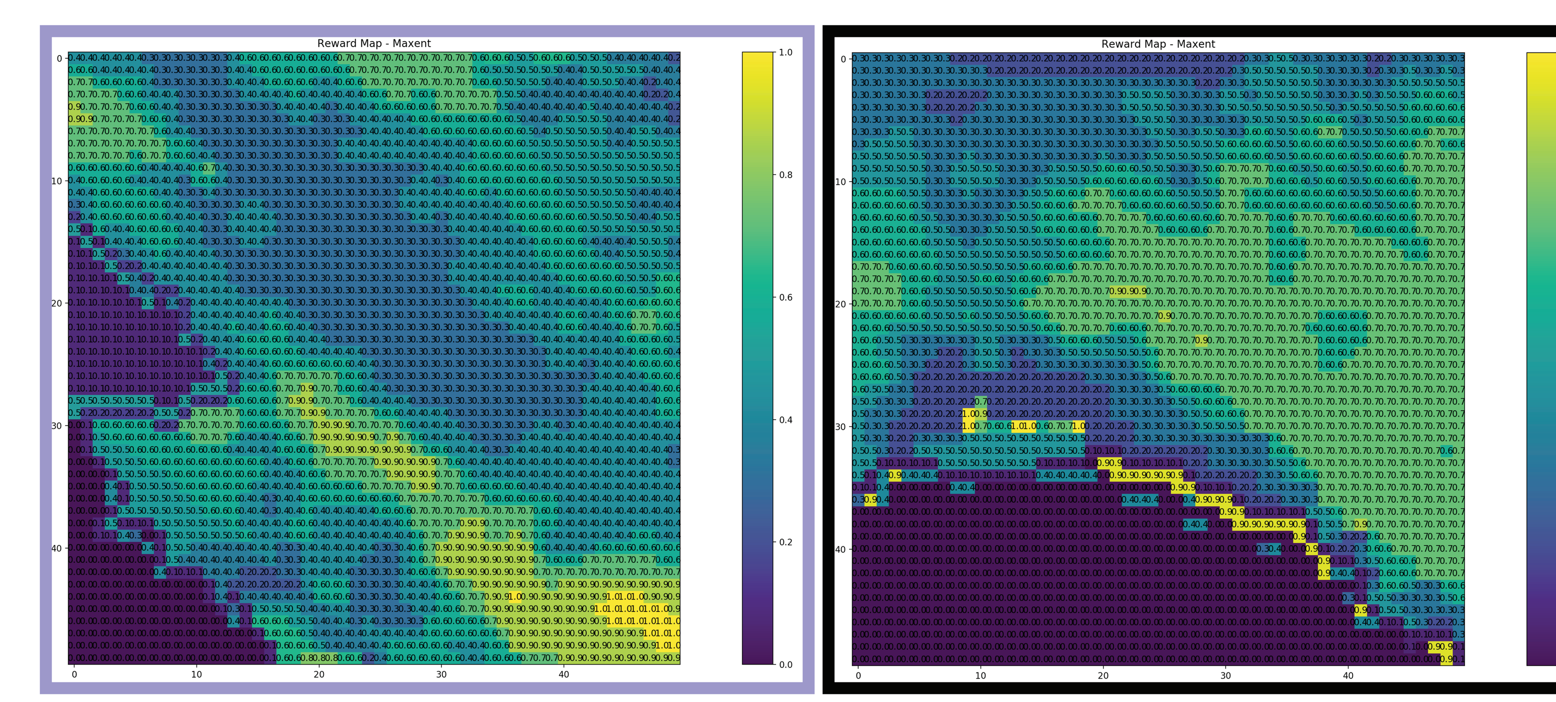
Because of the computational complexity of our model, it has a long runtime. Without using a supercomputer or parallel processing. Before investing time and resources, we first trained and fit our model on a simplified data set over a smaller subsection of the geographic area. Figure 4 illustrates the results from select subsections.

When testing the accuracy of the subset outputs, we found the following hyperparameters to be successul. Those marked with [1] were continued values from the GitHub implementation, in conjunction with our research: learning rates (0.02)[1], error values (0.1)[1], and gamma: 0.5-0.8.

We lowered the minimum accepted Gamma value from 0.8 to 0.5 because we needed to balance speed, efficiency, and accuracy. Consequently, our subsection size, 50x50, was the maximum size we could run over a high traffic area in a reasonable time (~12 hours).

For our the probability distribution we used a maximum entropy approach. The probability distribution does not show preferences other than feature expectations; with maximum entropy, equivalent rewards have equal prob while higher rewards are exponentially preferred.

Figure 4) *Visutalization of generated reward maps; select 50x50 square samples of the algorithm's learned output.*



Model

With our model we are attempting to create an educated path-interpolation algorithm that can adapt to any animal as an agent. This would require the respective area's enviornment data for feature collection. Our CLI includes the option for loading and parsing such input files through command line flags. Our machine learning algorithm uses an inverse reinforcement learning (IRL) model.[1] The foundational idea to reinforcement learning (RL) is to, "place an agent in an environment and reward or punish their actions. Over time, this results in an agent that acts in a way to maximizethe reward we specified." [2] Inverse reinforcement learning instead tries to learn the reward function, given policy and information about how the agent moves in the enviornment through its actions.

Future Steps

1) Parallelizing the code to obstain a reasonable computation time for running the code over the entire dataset

2) Use an IRL Deep Learning Model for potentially superior results

3) Adding in optional support for time dependent feature data, for example termperature, humidity, & pressure data

IRL & RL Vocabulary Terms Defined

RL Goal: learn policy to determine actions based on maximizing returns: $\underset{\pi}{argmax} \underset{\tau\sim\pi}{E}[R(\tau)]$

Agent - model that acts and learns within our environment

Environment - the limited space that the agent exists within that defines rules of what is possible

State - relevant information about environment with respect to agent's actions and results

$$s_{t+1} = f(s_t, a_t)$$

Actions (a_t) - agent can perform actions that alter the state

Policy - the set of rules an agent follows to determine action it should take given its observations about the environment ($\pi : s \to a$)

Returns - environment provides returned value to agent after it takes an action (reward or punishment)

$$R(\tau) = \sum_{t=0}^{T} \gamma^t r_t$$

[1] Maximum Entropy Inverse Reinforcement Learning: https://www.aaai.org/Papers/AAAI/2008/AAAI08-227.pdf | [1] Implementation from: https://github.com/yrlu/irl-imitation | [2] CSCI470 (Fall 2019) Lecture 22 - Reinforcement Learning