Satellite Image Classification: Progress Report

Anthony Cannistra
tonycan@uw.edu

Abstract

The purpose of this document is to elucidate my progress toward the CSE546 final project. In summary, we show herein the infeasibility of the originally-proposed project, the exploration phase of the first pivot away from the original project, the infeasibility of the second proposed project, and current newest work on the final settled project. As proposed, this project intended to use a neural network approach to classify pixels within a satellite image as `snow` or `no-snow` for the purpose of identifying snow accumulation and melting, with applications in watershed science and climate change research.

1 Pivot 1: Snow to Toxic Algae Blooms

This project is my first foray into using remotely-sensed data for any purpose. As a result, I've spoken with members of my lab with experience and have read to get a better idea of the data and its applications. In these conversations it became apparent that the identification of snow pixels is a rather intractable challenge, and has been attempted by more people than I originally expected. In addition, the technology onboard satellites (especially NASA's MODIS satellite) is far more adept at identifying snow cover than my initial research indicated [Maurer et al., 2003].

There is still objective value in re-visiting this task from a machine-learning perspective, since the $(250, 250)$-meter pixels of MODIS are limiting in their spatial (and temporal) resolution, and other higher-resolution imagers (like Sentinel-2 or Planet Flock) have no built-in snow identification. However, I was effectively dissuaded by experts in favor of another more ecologically-motivated project.

This new project involves the classification of toxic cyanobacterial and algal blooms in Washington lakes using remotely-sensed imagery. It is similar to the snow classification task in that training and validation data come from terrestrial sources—the Washington Department of Ecology tracks algae blooms across over 150 lakes in Washington—which makes for a readily-available training set. This work is inspired by a talk a friend of mine attended, wherein this evaluation was completed manually [Ignatius et al., 2017].

2 Roadblock: Satellite Imagery Is Difficult

Thankfully, the decision to pivot projects came only a day or two after submitting the proposal, so I had ample time develop a pipeline for handling these very large data (despite Prof. Jamieson's staunch demand that the data be already usable—I wanted to take this opportunity to learn this pipeline, and figured I could do it). The pipeline begins with the Washington State Department of Ecology database of lake cyanobacterial samples and a geodatabase of lake polygons. I joined these two sets to create a geodatabase of cyanobacterial concentrations keyed on `(lake-polygon, measurement-date)`, which was then fed into an image acquisition API from the European Space Administration/Copernicus to extract imagery of each lake within 10 days of the WA Ecology measurement date from the Sentinel-2 satellites. It has been developed in Python, using `rasterio` for raster processing, `scikit-image` and `numpy` for the image wrangling, and `geopandas` for the

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

geospatial components. Upon inspection, I discovered more problems than I have time to deal with, and only one or two of which are related to machine learning. As they say, "a picture is worth 1000 words:"

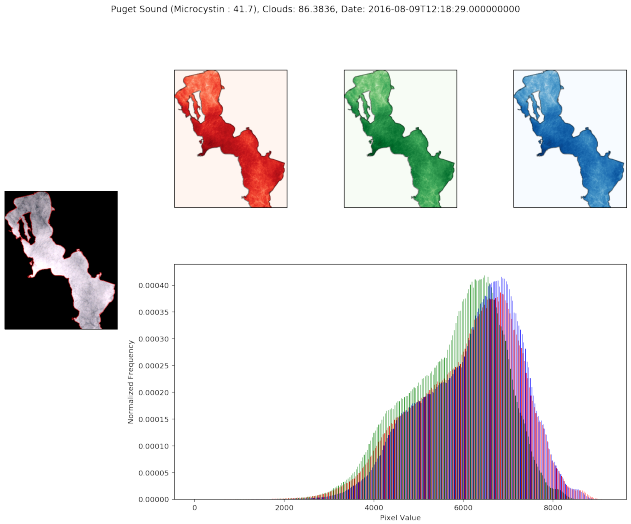

Figure 1: A Puget Sound algae bloom, covered by clouds. Left is the RGB Sentinel-2 image, top panel is each band, and bottom panel is combined uint16 histogram. In the title my regression variable (`Microcystin`'s) value of 41.7 indicates an algae bloom in the lake.

This is a beautiful image of an algae bloom in a section of Puget Sound, completely obscured by clouds. Almost all of the images I extracted had clouds covering the lake—I'm not sure why I didn't expect this. Here's another example of a problem:

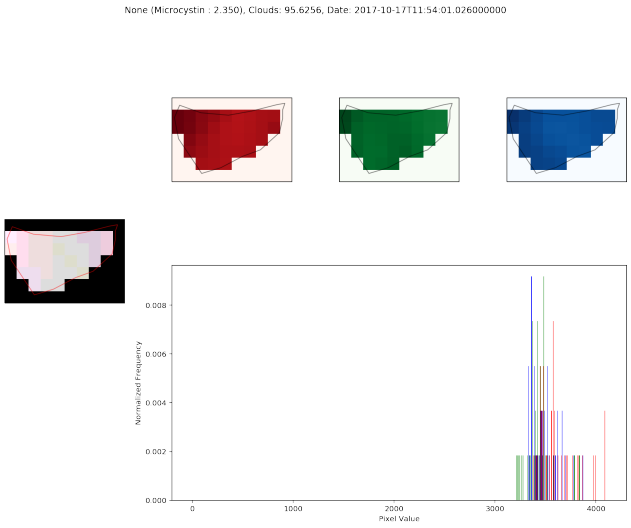

Figure 2: An unnamed miniscule lake algae bloom, covered by clouds. Left is the RGB Sentinel-2 image, top panel is each band, and bottom panel is combined uint16 histogram. In the title my regression variable (`Microcystin`'s) value of 2.35 indicates an algae bloom in the lake.

This time we have a lake which consists of no more than 20 pixels (which also happens to be covered by clouds, as seen in the high-intensity values in the histogram). **Herein lies an interesting ML problem I had hoped to tackle:** how to convert very disparate, heterogeneous image types into a feature set which can be understood by a machine learning model? Unlike other image processing

tasks which require standardized inputs, this particular task has *lakes* as input, which creates an interesting opportunity to evaluate different approaches in homogenizing the inputs.

There are methods to attempt to tackle the problem of clouds in these images, as well as to handle the homogenizing of inputs. I don't have the time to deal with them for this project. I mistakenly thought the data I had access to would be usable in this context, and it simply isn't. As a result, I will be shifting the project still within the satellite domain but to a much more machine-learning-focused task intended to enhance my understanding of the power of kernel-based linear methods.

3 Pivot 2: Kernel-Based Linear Methods for Satellite Image Scene Classification

Because I'm interested in Prof. Jaimeson's claims that linear methods can perform as well if not better than neural networks on the majority of learning tasks, I'm curious about stepping away from the neural network hysteria and using linear classification techniques that we've studied in class to classify satellite scenes. This project is appealing because it utilizes an already-existing dataset of labeled satellite images of homogeneous dimensionality from a Kaggle competition (`https://www.kaggle.com/c/planet-understanding-the-amazon-from-space`), and because there are many alternative and neural-network approaches which have been evaluated in a standard framework to which I can compare my results.

In service of this project I've downloaded the data (~30GB) and have begun an exploratory data analysis using AWS hardware. I will compare the performance of kernel multi-logit regression and multi-class support vector machines, which I expect to be very similar on these data. I'm also quite curious whether any regularization will have an impact on the results, given the dependence on pixel values to each other. I am implementing these methods using `sklearn.linear_model.SGDClassifier`, which is a flexible SGD implementation which allows me to perform minibatch training, since these images will not all fit into memory. I will also explore whether singular value decomposition can expose any interesting underlying features of this high-dimensional $[d = (256, 256) = 65536]$ data. These results will be compared to the highest-performing method in the Kaggle competition to demonstrate the relative performance of linear methods to the neural network approaches taken by the competitors.

References

A. R. Ignatius, T. Purucker, K. L. Wolfe, M. O. Galvin, B. A. Schaeffer, and J. M. Johnston. Spatial analysis of freshwater lake cyanobacteria blooms, 2008-2011. 2017. Ecological Society of America Annual Meeting, Portland, OR.

E. P. Maurer, J. D. Rhoads, R. O. Dubayah, and D. P. Lettenmaier. Evaluation of the snow-covered area data product from MODIS. *Hydrological Processes*, 17:59–71, Jan. 2003. doi: 10.1002/hyp.1193.

Appendix

For posterity's sake, I'll show some examples of *good* training data that I acquired for the algae classification task which I will likely complete at some other point.

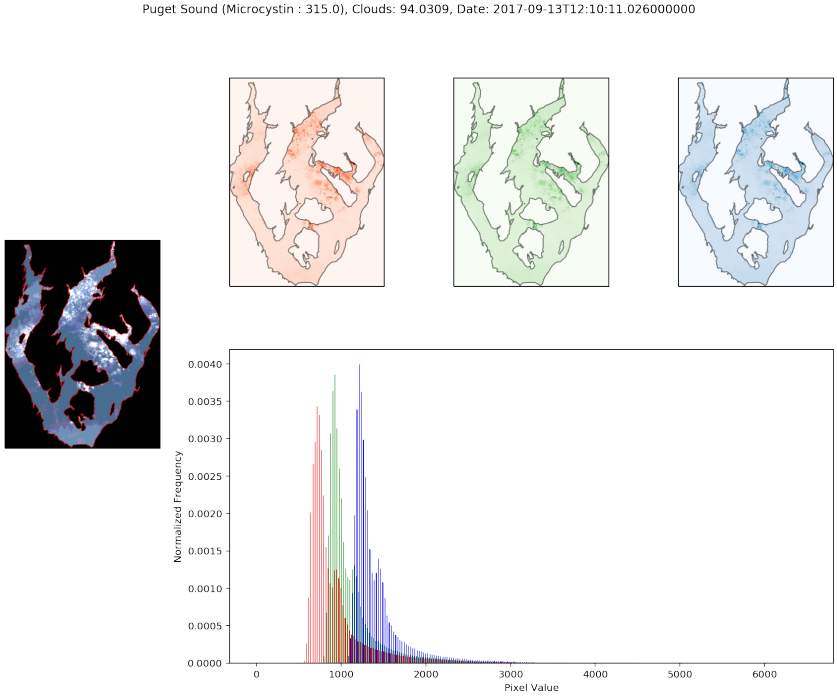

Figure 3: Puget Sound with Few Clouds and a likely algae bloom.

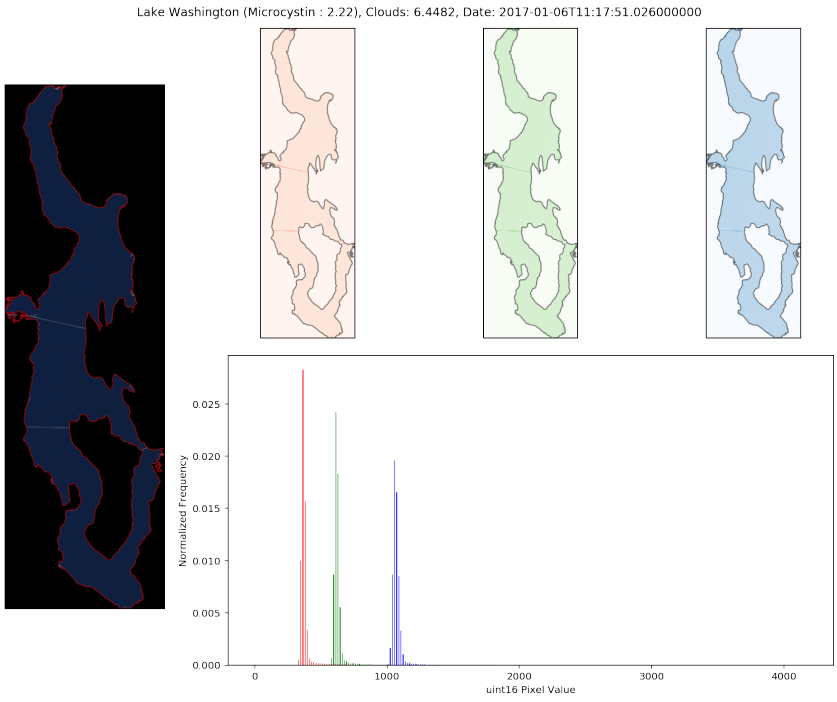

Figure 4: Lake Washington with No Clouds and no algae bloom.