ML & Climate | Final Paper

Angela Zhang | az2542

May 12, 2022

1 Abstract

Data sparsity is one of the biggest issues facing the machine learning community. Many techniques for generating synthetic data are often utilized to help make up for small or incomplete data sets. Alternatively, we would like to explore the options for combining two sources of data to create one dataset that can be analyzed with machine learning techniques. For climate related data in particular, there are many existing sources of data which are formatted and indexed by latitude and longitude. In this paper, we aim to explore the efficacy of combining datasets on common spatial information for sea ice thickness and sea surface temperature to evaluate whether any productive analysis can be made.

2 Introduction

Sea ice, which is simply frozen ocean water, exists mainly in the Arctic and Antarctic. It generally grows during the winter months and melts during other times of the year, experiencing significant losses in surface area. Although many of us many not have encountered sea ice before, it is crucial to the global climate because it helps to deflect sunlight back into the atmosphere so that the areas covered by sea ice don't absorb as much solar energy, which could cause an increase in temperatures in the Polar areas. Great fluctuations in sea ice are harmful for the environment and can greatly affect ocean currents as well (1).

Likewise, sea surface temperature (SST) also has a large impact on the global weather and climate. Changes in SST have been linked to the El Nino phenomenon as well as major changes in rainfall and drought patterns.

3 Background

Both the thickness dataset and the SST dataset come from the Copernicus Climate Data Store. Sea ice thickness is provided in meters and the sea surface temperature is provided in Kelvin (3).

4 Methodology and Challenges

I first started with exploring the sea thickness dataset, which was given as a set of coordinates (including a 2D array for latitude and longitude) as well as associated variables (ice thickness) for specific coordinates. Initially surveying the data in NetCDF format was difficult because this was a completely new form of data. I wasn't completely sure what gridded data meant or how it should be processed and used. This misunderstanding led me to attempt to unpack all of the latitude and longitude values, as well as associated sea ice thickness into a csv, with each row representing one latitude, longitude pair. Because the latitude longitude grid was 432x432 in dimension, this process took quite a while for just an initial processing phase. What this did help reveal however, was that most of the thickness values were listed as N/A, which makes sense because sea ice probably doesn't cover everything in a square grid- there are bound to be land interruptions.

The run time for csv creation was exacerbated in the second dataset, where there were 720 given latitudes and 1440 longitudes. Realizing that this was not a feasible options, I turned to other options to try and process this data and discovered that people generally use the xarray python package to assist with manipulation of NetCDF data. xarray was written specifically to deal with climate data and includes helpful operations such as moving from datasets (an xarray data structure) to dataframes and vice versa. Converting our data to a dataframe allowed us to drop all N/A values as well. This does have a potential of skewing prediction towards never having an n/a value, but I think there are probably ways to add back in n/a values given certain knowledge of where there is and isn't sea ice. I thought initially dropping N/A values might help with reducing down the amount of rows we had on the dataframe.

A lot of time was spend experimenting with xarray. One particular point of confusion was the format of the data provided for the sea ice thickness. The actual thickness array itself included values for an x and y pair. This x and y pair was meant to be used to access the latitude and longitude grid. In comparison to the SST data, where latitude and longitude are directly offered as variables as well, this was more difficult to figure out how to process. The dimensions between these two in terms of grid space provided was very different as well and poses another challenge for data merging.

After moving on to processing the SST data, it seemed that the techniques used for processing the thickness data would not work. At first I thought this might be due to having more latitude and longitude values, but after taking certain slices of these values and re-indexing for a smaller dataset, I was still running into memory errors. Eventually I realized that the thickness dataset includes values such as gradient fields, SST standard deviation. I removed these for the sake of our immediate task. It became much easier to convert the dataset to a dataframe.

After this, this first thing I tried was to build a 1 nearest neighbor model using the thickness data as the training data. The goal then in the classification phase was to pair each of the points from the SST data with the "nearest neighbor". The distance metric was initially just geopy's distance which calculates the distance between latitude and longitude (4).

Another option was to explore regridding programs and algorithms (some of which use a nearest neighbor model as well). I started out with xESMF, an ESMF based regridding program, which worked really well for the SST data because latitude and longitude were included directly as variables (5). However, because of the aforementioned lat/lon issue with the thickness data, there was additional processing needed before we could build a regridder for the data.

I was successful in building a new dataset which would use latitude and longitude as variables do that a more direct association with lat/lon and thickness pair was formed. The new issue was that there were now many more latitude and longitudes for thickness. We were actually able to reduce down many of the SST data due to slicing specific sections of that data to more directly align with the given grid for sea ice thickness. However, now that thickness was the larger dataset, new memory issues arose when trying to convert to and from the xarray dataset, which was a vital step in being able to build a regridder which would also include time.

Unfortunately, this is where I ran out of time. Because I lack experience with data science in general, I think working with a completely new form of data was certainly a challenge and probably hindered progress more than it should have. I will discuss future works in the next section.

5 Results and Future Work

As there are not tangible results for this project yet, I can describe short term and long term goals for next steps.

In the immediate future, I want to perhaps to some reduction of the lat and lon dimensions for the SST data while it is still in a dataframe before trying to push this back into a dataset. Conversion between dataset and dataframe is where we are running into these memory issues for the thickness dataset. I think manually selecting both time and latitude dimensions based off

of the data from the SST dataset can help greatly reduce the amount of data and will therefore allow us to move onto the next step.

After that, I plan to regrid both datasets with the same range and increments, which in theory will provide us with identical datasets that differ only because one contains thickness values and the other contains SST temperatures.

Once the data is in this format, we will be able to perform a merge between the two dataframes and produce a dataset which contains both sea ice thickness and sea surface temperature information. This will allow much more room for performing machine learning analysis.

I think one important thing to consider is how we can evaluate whether the new merged dataset is viable to be used. A lot of regridding and interpolation can corrupt the meaning of the original data, thus rendering the merge useless. One idea might be to use two datasets with a known, proven relationship. After performing regridding and merging with these datasets, we can analyze whether the original relationship is preserved. Because sea ice thickness and sea surface temperature do not yet have a solidly proven relationship, I think using other dataset first might be helpful to verify the validity of these processing techniques.

6 References

1. https://nsidc.org/cryosphere/seaice/index.html
2. https://oceanservice.noaa.gov/facts/sealevelclimate.html
3. https://cds.climate.copernicus.eu/cdsapp!/home
4. https://geopy.readthedocs.io/en/stable/
5. https://xesmf.readthedocs.io/en/latest/why.html