Project 1: Artic Sea Ice Feature Detection

* The first project that I have been working on involves feature detection in arctic sea ice. The feature that we are focused on detecting is where Ice Cracks may develop.

Motivation:

* Our overarching goal of this project is to use the data given in order to develop a method to determine where possible Ice Cracks may form. To develop this method, the data that we were given is of movement of an ice sheet, which was found by the use of satellites passing over the ice sheet at different time periods. Given in the data was a gpid, which is the identify of part of the ice chunk, in order to more easily track movement over time. Additionally, the location of the gpids in terms of coordinates were given and labeled as x and y. We were given 22 days’ worth of data, so the observation time of when the satellite noted a gpids location was also given. For our clustering, we rounded this in order to just have days, which leads into the k column, which is the image index. Sometimes within one day the satellite may have passed over a region multiple times, so would have multiple observations on the same day.

Sea Ice Motion Animation:

* In this animation, where each frame is a day of data, each dot on the map is the location of a gpid. The arrow head is pointing in the direction that gpid is moving, so we can start to see that there is not a cohesive movement among the entire ice chunk, meaning that cracks are going to form in places. The dots where missing arrowhead are places where there is no location data on that day (so in other words missing data). As you can see the missing data tends to come in patches, due to the satellites.

Explanation of Problem:

* This plot shows the trajectories of each of the gpids over the 22 days. So each day we had information for a gpid, a dot was plotted, and then they were all connected with an arrow head at the end pointing in the direction of the movement. The colors don’t mean anything, I think they are just a way in order to see the different gpids better. Once again in this graph, you can start to see some of the differences in movement of the gpids. I will talk about this in more detail later, but we want to use this data in order to cluster similar gpid trajectories.

Challenges:

* Before I get into our proposed method for sea ice crack detection, there is a number of challenges with this problem and our data set. First, a common method for spatio-temporal clustering, is a density based approach (for example to cluster popular driving routes), however working with density will not work with our data due to how the gpids are laid out. They won’t really cross paths, and the same gpids will always be around each other. Secondly, as I mentioned before, when have missing data, we are missing it in patches, which makes it difficult to interpolate over. Therefore this leads to another challenge, typical interpolation methods are not suitable due to the non-smooth spatial process and the nonstationarity due to ice moving as patches. Finally, we only observed motion data, so don’t know much else about the ice chunk.

Proposed Method:

* On this slide I give an overview of our proposed method. First we wanted to cluster similar trajectories in order to identify patches of ice that were similar to each other. Secondly, we wanted to create polygon labels in order to get consistent group membership over time. And lastly, we wanted to be able to use space-time interpolation within each ice pack where a cluster was identified.

Clustering with Bounding Box:

* So since we do not have any other information about each gpid at a time point, we had to create some features in order to create a distance measurement for clustering. This is where Susan’s idea of a bounding box came in. The idea was that for each gpid trajectory, we could draw a bounding box around it, this way to the missingness of the data is accounted for without having to interpolate. So included in our bounding box was the min/max latitude and longitude in a time frame, length traveled (beginning to end of time frame) for latitude and longitude, and also the direction the trajectory moved. Additionally, to try and keep clusters together, we added in the average lat and long as well. We then used these features as inputs into Kmeans clustering, after first standardizing them. The number of clusters was determined using the silhouette statistic, though I do want to place with other measures determining cluster numbers. The idea is that the boundaries of each cluster would be where the ice cracks form, because have different movements going up against each other.

Clustering at 1 time point using all available days to create bounding box:

* This slide has one example of what the output would look like. In this example, we used all 22 days in order to create our bounding box. This also is just a picture of what everything look like at the first time point. Can create this same plot for the following days, and while the cluster membership won’t change, it does show the movement in the ice chunk. So, as you can see 8 different clusters were created (show example of where a crack may form). Also, the unfilled boxes are where the missing data is found. I tried many different things while working at this stage. We also have this built out, so can create the bounding box by week, so that we are clustering smaller trajectories, for a more detailed look. This is what is used in our interpolation method that we will talk about later on. But I also played with the inputs into the kmeans clustering algorithm and additionally tried hierarchical clustering until we landed on something we though gave good results. Could more things still be tried, yes.

Currently working on: Interpolation:

* At the moment, I am working on creating an interpolation technique for the missing x and y values of the gpids. However, there are a few difficulties with this. Once again, missing data in patches makes things difficult, because can’t rely on nearest neighbors. Additionally, when using spatial temporal interpolation, in order to calculate the distance matrix, you need latitude and longitude, which is what we are missing. Our method was to use polygon intersections. So here is where we used the clustering by week, so essentially we have three different sets of clustering. For each of these we created polygons of each of the clusters to show the spatial neighbors. Then we found the intersection with a different week (so say week 1 and week 2) in order to find temporal neighbors. Once we find, these spatial-temporal neighbors, want to create a grid, to have some reference for when missing gpid location, and be able to interpolate onto this grid (haven’t done yet).

Interpolation Example:

* On this slide you can see Weekly intersection of polygons to find the spatial-temporal neighbors. For example, this first plot is the intersection of week 1 and week 2 polygons. Some issues with the weekly polygons with overlapping due to the convexity of polygons, so some of these intersections are now show, because a different one on top of it. But kind of just moving forward for now to see if this is a problem that really needs to be solved in the future.

Next Steps: Finish Project 1:

* For this project, now that I have found the spatial-temporal neighbors, I need to use this to interpolate the missing points Additionally, in order to see if this method is even beneficial, should compare it to just using linear interpolation, as what was done previously. After this, need to create a pipeline so this can become a more automated process, if eventually get more days of data.

Next Steps: Project 2

* Project Susan has talked about. Don’t know too much details about it at this point, but the general idea of what I understand that I’ll be working on.