Clustering SST Data - Report

Lokesh Venkatesh
Applied Mathematical Methods

September 30, 2024

1 Methodology

- We first weight the raw SSTs using the cosine-weights generated. The Anomalies have already been generated for us in the notebook given.
- We then need to remove the NaN values from the dataset by just filling/replacing them with a value that would be very unlikely to be any of the other observed data points. Here, I've just used T = -1000.
- We then perform the clustering after appropriately reshaping the array before passing it with scikit-learn's KMeans function.
- After clustering the data, we reintroduce the NaN values back into the output based on which (lat,lon) coordinates originally had NaN values, implying that those were land regions.
- We then finally plot the contour plots for these clustered data. The number of clusters chosen, for Case 1: Raw SST Data and Case 2: Weighted Zero-Mean Anomalies, have both been chosen according to some tools used for optimisation, as explained in the below sections. The outputs for the plots may be found in the notebook submitted.
- Note that the number of clusters passed includes one cluster to be all of the NaN values, i.e the land regions. So the number of actual SST clusters is just n-1.

The overall conclusion from this exercise, even though the code was working fine and I was able to get clusters, was that K-Means is probably a terrible tool to cluster data with such diverse patterns. For example, even for n=2 sea clusters, I obtained 12 different outputs for the 12 times I ran the clustering procedure on the Anomalies.

2 Raw SST Data: Result & Discussion

I obtained n=5 clusters of SSTs on performing the clustering. These appear to form 5 bands from the equator all the way to the poles. This could just represent the different zones of sea surface temperatures in accordance with the overall climate of those latitudinal-regions of the globe.

The value n=6 (and therefore number of sea clusters=5) was obtained using something known as the Elbow method, which I found on this link: Elbow Method vs Silhouette Coefficient in Determining the Number of Clusters. I'd also used a library called yellowbrick, and a variant of the Elbow Method known as the Calinski-Harabasz metric to identify what would be an optimum 'n', and it turned out to be 6. The corresponding plot of this metric for different values of K can be found in the notebook. Note that I'd also tried plotting the Silhouette Analysis chart for n=6, even though it was of barely any use in identifying a useful value of n, since when it was plotted for different values of K, it just represented a monotonically decreasing curve.

3 Weighted Zero Mean Anomalies: Result & Discussion

The exact same procedure for coding the clustering process was followed for the Anomalies as for the SST, and this time I obtained n=3, after eliminating the trivial case of n=2. The latter turns out to be a trivial case since it simply puts all the land/NaN values in one cluster and all the sea anomalies in the other cluster, which is simply meaningless. As mentioned earlier in this report, the clustering returned different charts when run multiple times, and this is likely one of the biggest drawbacks of using K-Means on a dataset as large as this. Nevertheless, what I noticed on a few of the trials was a large cluster off of the western coast of the Continental Americas, spreading well across half the Pacific Ocean. One guess was that this could be representing the El Nino-La Nina phenomenon that occurs in this region once in a while. The corresponding plot can be found saved in the submitted notebook.

For this case, I again used the Calinski-Harabasz metric and the yellowbrick library to identify that n=2 was the most suited number of clusters, which was as explained in the previous paragraph, the trivial case. Thus, I chose n=3, which basically amounted to 2 sea anomalies' clusters.