Determining Best Clustering Technique Which Mimics Real World Visuals on WWLLN Lightning Data

Alejandro Herrera, Ben van Oostendorp, Aeon Williams

Dr. Barnabas Bede, Dr. Yilin Wu, Dr. Jeremy Thomas, Dr. Natalia Solorzano

DigiPen Institute of Technology

**Introduction**

There are many different techniques for clustering data, but the question on which method best illustrates what we see in the real world with lightning strikes originating from the same cloud/group of clouds was not fully explored. We chose 2 commonly used clustering techniques and one slightly less common for clustering lightning to inspect the differences and compare with what we would expect them to look like in the real world.

Keywords: clustering, lightning, kmeans, dbscan, fcm

**Data and Methodology**

The data used was obtained from the World Wide Lightning Location Network (WWLLN), which is lightning strokes from all over the world. The data was then processed by (honestly not sure how they get the storm centered coords but that’s what goes here). Each point of data now has a longitude, latitude, date and time, and distance from the center of the storm. From here, we did a lot of visualization of the data, by loading it up, selecting the data form every 30 minutes, and then began to cluster with K-Means, DBSCAN, and FCM. Each clustering algorithm was setup with data in two different ways: partition the data based on distance to the storm and bucketize them, then cluster within the buckets, and taking the whole data, 30 minutes worth, and running the algorithm.

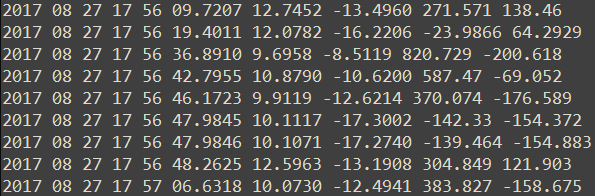


Figure : WWLLN Data with storm centered coordinates

**K-Means**

K-Means is a method of clustering that aims to partition n observations in k clusters, where each observation belongs to a cluster, and the center, or centroid, of the cluster is the mean of all points in the cluster. K-Means sets to minimize the variance of each cluster by optimizing the squared errors. It creates clusters based on *k* number of clusters.

**DBSCAN**

DBSCAN, or density based spatial clustering of applications with noise, is a density-based clustering algorithm that groups n observations based on density and closeness of observations in space. It creates clusters based on input parameters such as the maximum distance a point can be from a neighborhood and how many observations it requires within the neighborhood to be labelled a core observation. Observations that do not fit into the criteria are labelled as noise or outliers.

**Fuzzy C-Means**

Fuzzy C-Means (FCM) is very similar to K-Means in that it too aims to partition n observations in k clusters, but the membership of each observation to each cluster is not 1 (entirely belonging to) or 0 (does not belong). In FCM, observations can exist in multiple clusters to varying degrees, where a higher membership value means it exists more in the center of the cluster than observations with lower membership. The algorithm we used was developed by J.C. Dunn and improved by J.V Bezdek.

**Comparison of the Clustering Algorithms**

Once we ran each clustering algorithm on the data, it was noticeable that there were differences between each algorithm. We then looked at each of the algorithms pair wise to notice differences, compare results on the same data, and ultimately look to answer the question of which algorithm would best emulate the results we see in real clouds.

**K-Means vs DBSCAN**

**K-Means vs Fuzzy C-Means**

**Fuzzy C-Means vs DBSCAN**