Extracting Dense Regions From Hurricane Trajectory Data

Praveen Kumar Tripathi
University of Texas at Arlington
praveen.tripathi@mavs.uta.edu

Madhuri Debnath
University of Texas at Arlington
madhuri.debnath@mavs.uta.edu

Ramez Elmasri
University of Texas at Arlington
elmasri@cse.uta.edu

ABSTRACT

Weather data is a classic example of spatio-temporal data, with time and space as two of its key attributes. Clustering has been one of the key techniques used for analyzing the storm trajectories. Trajectory based clustering algorithms consider whole trajectories as clustering units, or in some cases the segments of the trajectory, i.e., sub-trajectories, are considered in order to capture local similarities among long trajectories. Our work takes a different approach, by considering a trajectory as a set of points, then focusing on the point data for finding the regions that are hot spots for the storms. We use DBSCAN algorithm, and consider spatial (*longitude*, *latitude*) as well as non-spatial attributes (viz., *wind speed* and *time*) for the similarity measure. The results show the impact of the respective non-spatial attributes on the spatial attributes during clustering and hence the identified dense regions. For the temporal analysis, we used a relative temporal framework by normalizing relative time stamp order in the trajectory by the length of the trajectory to consider storms of different lengths. We use quality measures to validate our clusters. Post processing on the obtained clusters identifies the regions from where the storms are more likely to originate, and the regions where the storms are most likely to land. Another useful result is the key regions that the storms are most likely to traverse.

1. INTRODUCTION

The abundance of spatio-temporal tracking data in applications like global positioning system (GPS), hurricane and storm tracking data and animal movement data have made their analysis very important. This analysis is vital in knowing and managing the traffic pattern of vehicles, monitoring and predicting weather conditions, examining wild animal behavior and movement as well as analyzing the spread of a disease. A number of attempts have been made in this domain to analyze these kinds of data sets. Some of these analysis could be found in [3, 4, 6–12]

For our task of identifying the dense regions of hurricane activity, we use a clustering algorithm called DBSCAN [5]. Clustering is a very useful task in data mining, which groups similar objects (physical or abstract) together [2]. The weather trajectory data has been analyzed using clustering algorithms in [6] and [7] (see Section 2).

Since the hurricane data is in the form of a trajectory, that represents the spatial locations of hurricane at different time instances, DBSCAN has been used which is a spatial clustering algorithm. For the current analysis we consider the hurricane data as point data, unlike the approach in [6] and [7]. This approach has been motivated by the need of the analysis and also by the fact that the hurricane trajectory lengths are different.

We cluster the data to obtain dense regions that effectively identify the hot spots for the storm activity. These dense regions have been identified considering different combinations of parameters. Initially we do only the spatial dense regions identification considering latitude and longitude, then we incorporate wind speed also as an additional attribute. This has been done to evaluate the impact of the non spatial attribute on the dense regions identification. Finally we do clustering considering the spatial as well as the temporal attribute to identify the spatio-temporal dense regions. For this analysis we consider the relative time framework as we are interested in the storm progression. We normalize the temporal value in the range of $[0-1]$, to handle the different length hurricanes. Our framework for combining the spatial and non-spatial attributes is inspired by the approach in [3].

We identified some dense regions that would be useful for the domain experts. First the locations from where the storms are most likely to originate, second the locations where the storms are most likely to land and finally, the regions that have been mostly affected by the storm activity. For the storm staring location identification, we cluster the initial portion of the hurricane, whereas, for the potential storm landing locations we cluster the last portion of the hurricane trajectory. The clustering considering the whole hurricane length data will find out the dense regions of high storm activity.

We have used the hurricane (Atlantic region) data set for 50 years from 1950 to 1999. The data set was obtained from [1]. The data has six attributes, which are latitude, longitude, time, wind speed, pressure, and status. The data has been sampled in the interval of 6 hours. The dataset has 15319 data points and 496 trajectories.

2. RELATED WORK

In this section, we give a brief overview of eight related

GeoRich'14, June 22-27 2014, Snowbird, UT, USA
 ACM 978-1-4503-2978-1/14/06 ...$15.00
http://dx.doi.org/10.1145/2619112.2619117.

articles [4, 6–12].

In [4] a non-parametric approach to spatial trajectory clustering, called DENTRAC (DENsity based TRAjectory Clustering) is proposed. DENTRAC uses the non-parametric density estimation technique. The post processing of the obtained spatial clusters is performed to get more domain specific knowledge.

The article [6] proposes a partition-and-group framework for clustering the trajectory data. Using the concept of minimum descriptive length (MDL) principle, most important points on the trajectory called characteristic points are identified. The original trajectories are now represented by connecting the consecutive characteristic points. Each segments thus obtained are called trajectory partitions. These trajectory partitions are then clustered using a modified version of DBSCAN algorithm, which clusters the line segments. Finally the clusters are represented by the representative trajectories.

In [7], a clustering algorithm is given for the trajectory data that uses the combination of techniques from data mining, computational geometry and string processing. The trajectories are pre-processed to remove noise after which they are segmented into sub-trajectories. These segments are then classified and accordingly labeled on their geometric properties e.g., "wide left right" or "short straight segments". The next phase of the algorithm finds the frequent occurring substrings; these are called the $motifs$. Algorithm then maps the sub trajectories corresponding to the motifs to some feature space. The next stage performs the density based clustering and the final stage does the post processing of the clusters.

In [8] a novel algorithm called Slicing-STS-Miner has been proposed for mining the sequential patterns from the spatio temporal data. This analysis is very valuable for analyzing the evolution of phenomena in spatial and temporal domain.

A sptio temporal pattern called convoy has been proposed in [9]. In this article authors propose various efficient algorithms for the convoy detection.

In the article [10], the trajectory clustering technique of [6] has been extended for trajectory classification. In this article two levels of clustering; namely, the region-based and trajectory based clustering is done. Clustering is used to find the discriminative features for classification. The first level of the clustering is region level which identifies the higher level, region based features of the trajectories. The second level of the clustering identifies the lower level movement based features. These two clustering collaboratively identify the high-quality features for the classification.

In [11] authors propose a classification technique for the trajectory data which incorporates the duration of the trajectory as an important feature.

In [12], a similar technique to [9] for mining spatio temporal pattern called the flocking behavior is proposed. The flock refers to the set of the trajectories that remain close to each other for some reasonable time interval. In the flock pattern mining both the time as well as the spatial attributes are required.

3. TRAJECTORY CLUSTERING ON POINT DATA

In our approach we have considered the trajectory data set as consisting of just the individual data points in the trajectories. In this analysis we are not constraining these points to belong to their respective parent trajectory or sub trajectory by enforcing them to belong to a line segment or a sequence of line segments as has been done in [6] and [7]. Since the core algorithm behind our analysis is DBSCAN [5] we review that algorithm first.

3.1 DBSCAN

DBSCAN is a density based clustering algorithm. It has two important parameters called the Min_{Pts} and Min_{Dst}. These two parameters determine the density of the data to be clustered. For a point to be evaluated as dense, we need to look at a neighborhood of size Min_{Dst} centered around it. In this neighborhood there should be at least Min_{Pts} number of data points to make this particular data item dense. On the basis of the density of the data points in the data set, DBSCAN identifies three types of points viz., 1)$core\ points$, 2)$boundary\ points$ and 3)$noise\ points$. Figure 1 gives a scenario of the data points and distinguishes between the three kinds of data points. Formally, these points are defined on the basis of $Min_{Pts} - neighborhood$, viz., $(N_{Min_{Pts}}(p))$ for a point p in the dataset D.

$$N_{Min_{Pts}}(p) = (a | a \in D \ and \ dist\,(p,a) \leq Min_{Dst}) \quad (1)$$

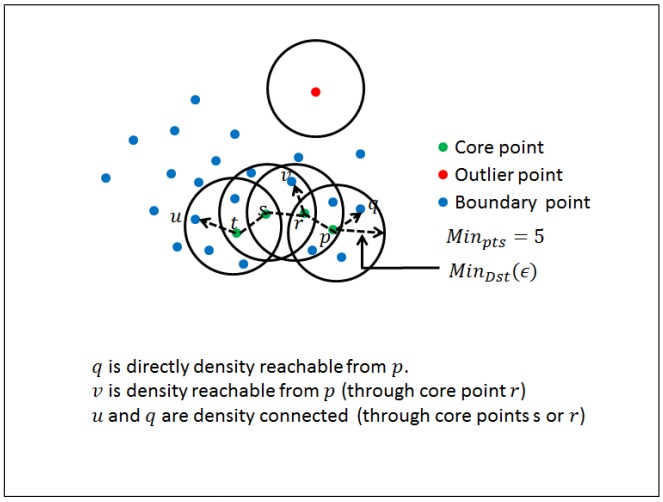

Figure 1: DBSCAN Types of data points: Core points, Boundary points and Outliers

Algorithm 1 The DBSCAN Algorithm

```
DBSCAN(D, Min_Dst, Min_Pts)
C = 0
for all unvisited point P in dataset D do
  mark P as visited
  Neighbor_Pts = regionQuery(P, Min_Dst)
  if sizeof(Neighbor_Pts) < Min_Pts then
    mark P as NOISE
  else
    C = nextcluster
    expandCluster(P, Neighbor_Pts, C, Min_Dst, Min_Pts)
  end if
end for
```

For every point p in the dataset D, its $Min_{Pts} - neighborhood$,

viz., $(N_{Min_{Pts}}(p))$ is determined on the basis of the parameter Min_{Dst} and similarity measure viz., $dist\,(p, a)$ (for example, Euclidean distance), between the point and its neighbors. If the size of $N_{Min_{Pts}}(p)$ for a particular point p, is not less than Min_{Pts} then the point is considered a core point. If the point p is not core but it lies in the $N_{Min_{Pts}}(q)$ of a core point q, then it is called a boundary point. If it is not a core point and also does not lie in the neighborhood of any core point, then it is called an outlier (see Figure 1).

To define the clusters in terms of DBSCAN, three more concepts have been defined, these are:
1) *directly density reachable*, 2) *density reachable* and 3) *density connected*. A point q will be directly density reachable only from a core point (p), only when it lies in the $N_{Min_{Pts}}(p)$. For example, point q is directly density reachable from the core point p in Figure 1. Similarly a point t would be density reachable from a core point p, if there is a sequence of data points $\{x_1, x_2, \ldots, x_n | x_i \in D\}$, where x_i is directly density reachable from x_{i-1}, and also $x_1 = p$, whereas $x_n = t$. For example point v is density reachable from core point p in Figure 1. Similarly the density connectivity between two points a and b in the data set is defined as the existence of a core point c such that the points a and b are density reachable from c. For example in Figure 1, points q and u are density connected with respect to the core point s (also r).

A cluster C is defined as the subset of objects satisfying two criteria: 1) Connected: means that $\forall p, q \in C$, p and q are density connected, 2) Maximal: It means that $\forall p, q$, if $p \in C$ and q is density reachable from p, then $q \in C$.
The structure of the DBSCAN is given in Algorithm 1, Algorithm 2 and Algorithm 3.

Algorithm 2 expandCluster

expandCluster$(P, Neighbor_{Pts}, C, Min_{Dst}, Min_{Pts})$
add P to cluster C
for all each point P' in $Neighbor_{Pts}$ **do**
  if P' is not visited **then**
    mark P' as *visited*
    $Neighbor_{Pts'} = regionQuery(P', Min_{Dst})$
    if $sizeof(Neighbor_{Pts'}) >= Min_{Pts}$ **then**
      $Neighbor_{Pts}$ = $Neighbor_{Pts}$ joined with $Neighbor_{Pts'}$
    end if
  end if
  if P' is not yet member of any cluster **then**
    add P' to cluster C
  end if
end for

3.2 Dense region extraction

Our contribution in this article is in analysis of the hurricane data to find the regions of high storm activity. We used DBSCAN algorithm which uses the parameter Min_{Dst} for finding the neighborhood and hence the density of the data points. Since the hurricane data set which has been used in this paper has spatial as well as non-spatial attributes, first we find the neighborhood considering only the spatial neighborhood, then we considered a non-spatial attribute (wind speed) also.

Algorithm 3 regionQuery

regionQuery(P, Min_{Dst})
return all points within $P's$ Min_{Dst} − *neighborhood* $(including P)$

3.2.1 Spatial Clustering

For the spatial clustering we considered the latitude and the longitude values of the data points. We considered the $Haversine\ formula$ for computing the distance between the two data points represented by their latitude and longitude values, i.e., $P_i = (\phi_i, \lambda_i)$, where ϕ_i is the latitude and λ_i is the longitude of the data point P_i. If we have two points $P_1 = (\phi_1, \lambda_1)$ and $P_2 = (\phi_2, \lambda_2)$, the $Haversine\ formula$ is given as:

$$a = \sin(\Delta/2) + \cos(\phi_1) \times \cos(\phi_2) \times \sin^2(\Delta\lambda/2)$$
$$c = 2 \times \arctan 2(\sqrt{a}, \sqrt{(1-a)})$$
$$d = R \times c$$

where $\Delta(\phi) = \phi_2 - \phi_1$ and $\Delta(\lambda) = \lambda_2 - \lambda_1$, $R = 6371Km$ is the radius of the Earth, and d is the Haversine distance. Since the spatial co-ordinates in geographical data sets are reported in terms of the latitude and longitude, it will be natural to adopt some distance measure that computes the circular distance between two points lying on a spherical object (earth).

3.2.2 Spatial and non spatial clustering

We extend the DBSCAN algorithm to handle the non spatial attributes also for clustering. Our approach to this analysis is inspired by [3], where authors extended the DBSCAN algorithm to incorporate non-spatial attributes. Let there be a data set $D = \{d_1, d_2, \ldots, d_n\}$, where $d_i = (x_i, y_i, a_i, b_i)$. Let (x_i, y_i) be the spatial attributes and (a_i, b_i) the non spatial attributes. According to [3], we can consider the spatial as well as the non spatial distances to find out the respective neighborhoods. More formally, let us consider the distance between two data points $d_1 = (x_1, y_1, a_1, b_1)$ and $d_2 = (x_2, y_2, a_2, b_2)$. Now if we denote the non-spatial distance as $dist_s$ between d_1 and d_2, then $dist_s(d_1, d_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$. Similarly, we can find the non-spatial distance between the same data points d_1 and d_2 as $dist_{ns}(d_1, d_2) = \sqrt{(a_1 - a_2)^2 + (b_1 - b_2)^2}$. Here we assume that the Euclidean distance is meaningful between the data points in spatial as well as non spatial domain. Let us define two user specified spatial and non-spatial threshold as ϵ_s and ϵ_{ns}, respectively. Now we can define two neighborhoods of a point d_k, which are the spatial neighborhood and the non-spatial neighborhood. The spatial neighborhood is $N_{\epsilon_s}(k) = \{d_j \in D | dist_s(d_k, d_j) \leq \epsilon_s\}$, and the non-spatial neighborhood is $N_{\epsilon_{ns}}(k) = \{d_j \in D | dist_s(d_k, d_j) \leq \epsilon_{ns}\}$. Finally the composite neighborhood will include the data points common in $N_{\epsilon_s}(k)$ and $N_{\epsilon_{ns}}(k)$ which is $N(k) = N_{\epsilon_s}(k) \cap N_{\epsilon_{ns}}(k)$. Effectively, the neighborhood $N(k)$ consists of the data points around d_k, which are closer to it with respect to spatial as well as non-spatial distance, given the respective thresholds ϵ_s and ϵ_{ns}.

For the non-spatial attribute we have used wind speed in our analysis, because wind speed is an important attribute of hurricanes. Further, only the wind speed data is available in the data set for all the records.

3.2.3 Spatio temporal clustering

Since the hurricane data is naturally spatio-temporal data, we need to consider the time also in the analysis. In the current work we have considered time also as a non-spatial attribute. We use the time in the following formulation. Let the user specified temporal threshold be ϵ_t. Here the data point $d_i = (\phi_i, \lambda_i, t_i)$, where ϕ_i is the latitude, λ_i is the longitude, and t_i is the time. $dist_s$ is the $Haversine$ distance, where as the $dist_{ns}$ is simply the Manhattan distance, viz., $dist_{ns}(d_i, d_j) = |t_i - t_j|$.

In the hurricane data set, hurricanes have been sampled at 6 hours intervals, so a trajectory of a particular hurricane of length l is represented as $Tr_i = \{(\phi_{i1}, \lambda_{i1}, 1), (\phi_{i2}, \lambda_{i2}, 2) \ldots (\phi_{il}, \lambda_{il}, l)\}$. In this work we have considered the relative time framework, where instead of using the absolute time in the hurricane tracks, we consider the relative time from the start of a particular track. This framework is more important in the current analysis because our aim is to analyze this data from the hurricane's movement patterns point of view. Since the hurricanes are of different length, we have normalized the time component by the length of the hurricane trajectory. Which is:

$$Trn_i = \left\{ \left(\phi_{i1}, \lambda_{i1}, \frac{0}{l-1}\right), \left(\phi_{i2}, \lambda_{i2}, \frac{1}{l-1}\right) \ldots (\phi_{il}, \lambda_{il}, 1) \right\}$$

Because of this normalization the time component of the trajectory will vary from $(0-1.0)$. Please note that the lower value of t_i close to 0.00 will signify that the trajectory data point is in its initial stage, where as the higher values close to 1.00 would signify that the particular data point lies towards the end of the hurricane. Similarly the values of t_i close to 0.5 will signify the middle data points in the trajectory. If we use the spatial and this temporal information together in the DBSCAN for finding the clusters then we will obtain spatio-temporal clusters.

4. EXPERIMENTS AND ANALYSIS

We have done the implementation and analysis using MATLAB on the hurricane data set described at the end of Section 1.

4.1 Analysing spatial attributes

Fig 2 shows the result of the DBSCAN algorithm applied on the 50 years storm data. For this experiment the parameter values were $Min_{Dst} = 35$ and $Min_{Pts} = 10$. We obtained 15 clusters of different sizes across the region. This choice of the parameters Min_{Dst} and Min_{Pts} is based on the experimentation. We tried to get the values of these parameters on the basis of the suggestions in [5], but we got just one single cluster as the output. Therefore, we ran the DBSCAN algorithm for different combinations of Min_{Dst} and Min_{Pts} and picked the above value for the case where we got well separated and compact clusters.

In order to analyze the results depicted in Fig 2 we will utilize Table 1, which summarizes certain properties of the 15 clusters. The big cluster in the center (Magenta stars) shows the most dominant region of the storms. Numerical values corresponding to this cluster, which has cluster id 4, can be obtained from Table 1. This cluster with cluster id 4 is ranked first with respect to the number of trajectories($Storm_{rank}$ (#traject.)) as well as the number of data points belonging to it viz., ($Storm_{rank}$ (#data points)). The number of different storm trajectories that pass through this cluster is 185 and it has total of 1673 storm data points (Table 1). We conclude that this region is the most prominent region in terms of hurricane storm activity. The next dominant region corresponds to the cluster with cluster id 3 (green stars), which has the second rank in terms of the number of storm trajectories (viz., 80) (Table 1) that passed through it and the number of storm data points (viz., 471) in it. On the other side, we have some smaller size clusters that are lowest in the ranks. Clusters with id 10 and 14 rank as the last two in terms of the number of storms that pass through them, where as the storms with id 12 and 13 rank as the lowest two in terms of the ranking on the basis of the storms data points that belong to them. We have some medium class storm clusters also. This kind of analysis will identify spatially the regions that are susceptible to storms, and at the same time we can identify relative susceptibility among them.

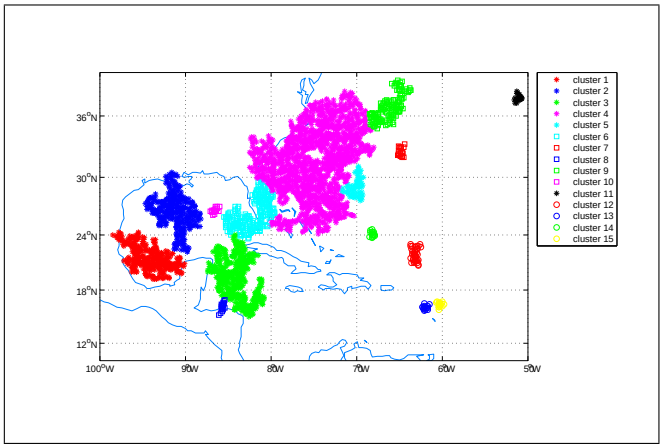

Figure 2: Storm Clustering with spatial attributes

Table 2: Storm Clustering analysis, Impact of Non spatial attribute $Min_{Pts} = 10$, $Min_{Dst} = 35$

$Min_{DstWspeed}$	$Mean(Std(Cluster\_i))$	(#clusters)
20	13.5669	5
30	16.222	6
40	22.4366	9
50	23.2284	9
60	25.458	12
70	25.9649	14
80	26.3079	14
90	27.6229	15
100	27.6691	15

4.2 Qualitative analysis of clusters

In this work we have extended DBSCAN algorithm to find out the dense regions in hurricane point data while considering the non-spatial attributes along with the natural spatial attribute. The two non-spatial attributes considered in this work are wind speed and time. We varied the wind speed threshold $Min_{DstWspeed}$ from $20 - 100$ (interval of 10), and the temporal threshold ϵ_t from $0.2-1.0$ in the interval of 0.1. Since we got a range of clustering results when we changed these parameters, we needed a quality measure for the results of clustering. We used the quality measure proposed

Table 1: Storm clustering analysis, on Spatial clustering , $Min_{Pts} = 10$, $Min_{Dst} = 35$

Storm ID	$Storm_{rank}$ (#traject.)	$Storm_{rank}$ (#data points)	$\#traject.(Cluster_i)$	$\#DataPts.(Cluster_i)$	$Color(Symb.)$
1	6	4	53	361	Red (star)
2	3	3	68	371	Blue (star)
3	2	2	80	471	Green (star)
4	1	1	185	1673	Magenta (star)
5	7	7	37	76	Cyan (star)
6	5	5	58	244	Cyan (square)
7	10	9	16	24	Red (square)
8	4	6	60	144	Blue (square)
9	9	11	18	20	Green (square)
10	14	10	9	22	Magenta (square)
11	8	8	27	40	Black (star)
12	13	14	12	15	Red (circle)
13	11	15	14	15	Blue (circle)
14	15	12	9	20	Green (circle)
15	12	13	14	17	Yellow (circle)

in [6]. This measure is given below:

$$\sum_{i=1}^{num_{clus}} \left(\frac{1}{2C_i} \sum_{x \in C_i} \sum_{y \in C_i} dist(x,y)^2 \right) + \frac{1}{2|N|} \sum_{w \in N} \sum_{z \in N} dist(w,z)^2 \quad (2)$$

where, num_{clus} is the number of clusters, N is the set of noise points and C_i is the i^{th} cluster. This quality measure computes the sum of the square error (SSE), which means the smaller this value, the better will be the clustering result.

4.2.1 Analysing combination of spatial and non-spatial attributes

We did some additional analysis that would give us more concrete information in terms of the nature of the storms. The data set includes the attribute value wind speed, which is available for all the data points, and is one of the key characteristics of hurricanes. We used this as a non-spatial attribute and redefined the distance parameter in the DBSCAN algorithm to combine spatial and non spatial values. This approach is inspired by the work in article [3]. Now the clusters that we get are the spatial regions that are prone to similar kind of storms in terms of the wind speed. In the redefinition of the Min_{Pts}, we need to consider the neighbors as the data points which are closer to the particular data point with respect to the spatial distance as well as similar in wind speed values.

As expected when we constrained the neighborhood criteria by incorporating non-spatial attribute, viz., wind speed the number of clusters as well as the size of the clusters was reduced. This is because now the clusters represent the regions that were influenced by the same nature of storms. When we relaxed the similarity in the wind speed to be 100 the result degenerated to the case of totally spatial clustering, as the wind speed similarity had no impact. But as we reduced the value of wind speed similarity to lower values (70, 50 and 30 respectively), we got a smaller number of clusters and they got more compact. In order to reflect that extent of compactness we have a column in Table 2 $Mean(Std(Cluster\_i))$, viz., the mean of the standard deviation of the wind speed in the individual clusters for the particular choice of the wind speed similarity threshold. The standard deviation goes down as we reduce the value of wind similarity. This measure gives an intuitive measure of the compactness and the homogeneous nature of the clusters.

Now using the measure in Equation 2, since it has the notion of distance, we used two distances separately. First we used the spatial distance using the latitude and longitude of the data points, and second the difference in wind speed of two data points. We got the following performance in Table 3. The results shows that as we reduce the $Min_{DstWspeed}$ value from 100 to 20 the clustering result improved in terms of Q_{Wspeed} as well as $Q_{spatial}$. This result is obvious as the reduction in the threshold value $Min_{DstWspeed}$ forces more compact and homogeneous clusters.

Table 3: Qualitative measure of clustering results

$Min_{DstWspeed}(mph)$	$Q_{Wspeed}(mph)$	$Q_{spatial}(km)$
20	4.92E+04	3.35E+04
30	1.49E+05	8.62E+04
40	1.15E+06	2.56E+05
50	1.43E+06	2.96E+05
60	1.83E+06	3.46E+05
70	2.20E+06	4.11E+05
80	2.41E+06	4.73E+05
90	2.53E+06	5.18E+05
100	2.55E+06	5.18E+05

Table 4 shows the spatio temporal clustering result. Here we have done the clustering using the normalized time parameter along with the spatial attributes. We reduced the temporal threshold ϵ_t from 1.0 to 0.2 and found the performance of the clustering results to improve in terms of the quality Q_{ϵ_t} and $Q_{spatial}$. These two parameters denote the quality measure in terms of the temporal and spatial homogeneity of the clusters.

Note that although the lower values of the quality measure in Equation 2 signify better clustering result, these best performance results may not be what the end user wants. This is because the clustering concept is very subjective.

4.3 Analysing storm starting and landing information

One more analysis that we have done on the storm data is about the landing and the starting information for the storms. Here we first considered only the starting three data points corresponding to the first three time stamps, for all the storm trajectories. The DBSCAN algorithm was run on this data set. The resulting clusters give the regions from where the storms are most likely to start. Figure 3 shows the potential regions from where the storms may originate.

Table 4: Qualitative measure of clustering results

ϵ_t	Q_{ϵ_t}	$Q_{spatial}$
0.2	1.81E+00	1.13E+04
0.3	2.23E+01	9.30E+04
0.4	1.23E+02	3.04E+05
0.5	1.54E+02	3.54E+05
0.6	1.79E+02	4.11E+05
0.7	1.96E+02	5.05E+05
0.8	2.02E+02	5.18E+05
0.9	2.03E+02	5.19E+05
1.0	2.03E+06	5.19E+05

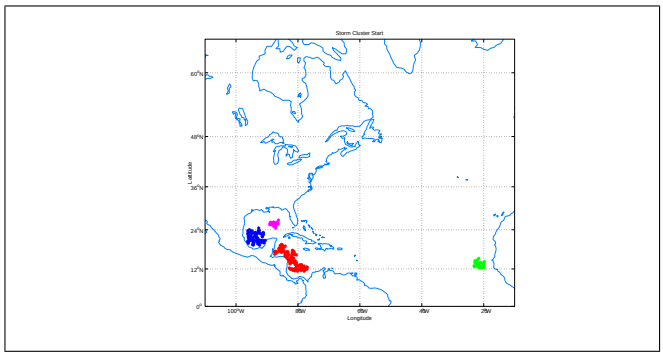

Figure 3: Storm starting clusters

Similarly we did the analysis on the last three data point of the storm trajectories. The result in Figure 4 shows the potential regions, where the storms may end.

Our motivation for doing the separate analysis for the landing and starting trajectory portion is based on the fact that when we consider all of the data points for the clustering using DBSCAN, these (landing and starting) information related with the storm activity may be lost as noise.

5. CONCLUSION

In this work we analyzed hurricane storm trajectory data to find areas of extreme hurricane density, as well as areas where hurricanes originate and land. We took a different approach to trajectory analysis by focusing on points along the trajectory, rather than line segments as in previous work. We used the DBSCAN algorithm for the clustering analysis. Initially the clusters are obtained on the basis of only the spatial attributes. After that, we looked at the influence of the non-spatial attributes, in particular wind speed, on the clusters obtained. We also propose a spatio temporal DBSCAN algorithm where the normalized relative time information with the point data has been considered as another non-spatial attribute for DBSCAN. We post processed the clustering results to obtain the storm starting, storm landing and storm tracking information. Our work differs from other work because of the focus on trajectory points, which results in identifying high-activity regions, as well as regions at start and end of storms.

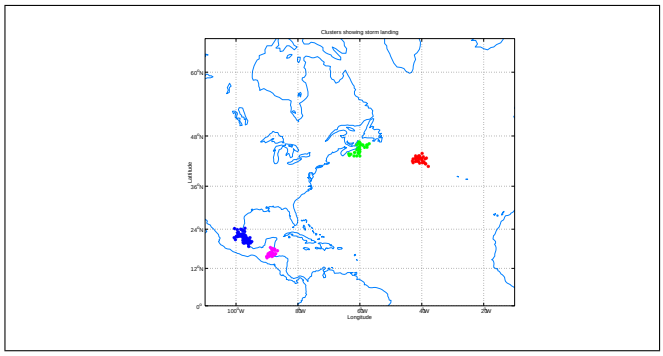

Figure 4: Storm Landing clusters

6. REFERENCES

[1] `http://weather.unisys.com/hurrricane/atlantic/index.html`.

[2] *Data Mining: Concepts and Techniques, 2nd ed.* Morgan Kaufmann, 2006.

[3] Derya Birant and Alp Kut. ST-DBSCAN: An algorithm for clustering spatial-temporal data. *Data Knowl. Eng.*, 60(1):208–221, January 2007.

[4] Chun-Sheng Chen, Christoph F. Eick, and NouhadJ. Rizk. Mining spatial trajectories using non-parametric density functions. In Petra Perner, editor, *Machine Learning and Data Mining in Pattern Recognition*, volume 6871 of *Lecture Notes in Computer Science*, pages 496–510. Springer Berlin Heidelberg, 2011.

[5] Martin Ester, Hans peter Kriegel, Jorg S, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. AAAI Press, 1996.

[6] Jae gil Lee and Jiawei Han. Trajectory clustering: A partition-and-group framework. In *In SIGMOD*, pages 593–604, 2007.

[7] Joachim Gudmundsson, Andreas Thom, and Jan Vahrenhold. Of motifs and goals: mining trajectory data. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '12, pages 129–138, New York, NY, USA, 2012. ACM.

[8] Yan Huang, Liqin Zhang, and Pusheng Zhang. A framework for mining sequential patterns from spatio-temporal event data sets. *IEEE Transactions on Knowledge and Data Engineering*, 20(4):433–448, 2008.

[9] Hoyoung Jeung, Man Lung Yiu, Xiaofang Zhou, Christian S. Jensen, and Heng Tao Shen. Discovery of convoys in trajectory databases. *Proc. VLDB Endow.*, 1(1):1068–1080, August 2008.

[10] Jae-Gil Lee, Jiawei Han, Xiaolei Li, and Hector Gonzalez. Traclass: Trajectory classification using hierarchical region-based and trajectory-based clustering. *Proc. VLDB Endow.*, 1(1):1081–1094, August 2008.

[11] Dhaval Patel, Chang Sheng, Wynne Hsu, and Mong Li Lee. Incorporating duration information for trajectory classification. In *Proceedings of the 2012 IEEE 28th International Conference on Data Engineering*, ICDE '12, pages 1132–1143, 2012.

[12] Marcos Vieira, Petko Bakalov, and Vassilis Tsotras. On-line discovery of flock patterns in spatio-temporal data, 2009.