# Storm Clustering for Data-driven Weather Forecasting

Xiang Li[1], Rahul Ramachandran, Sunil Movva and Sara Graves
University of Alabama in Huntsville

Beth Plale and Nithya Vijayakumar
University of Indiana in Bloomington

*In dynamic weather forecast paradigm, a weather model is launched in response to the storm events detected from real-time observation data. In such a situation, many storms may be detected and are normally clustered corresponding to several local storms. It is therefore more appropriate to launch weather models on spatial scales corresponding to clusters instead of the individual storms. In this study, we evaluate two clustering algorithms for their performance to cluster individual storms detected from real-time WSR-88D radar data. We also evaluate the performances of two statistical indices for determining the number of clusters in a storm data set. Based on this research, a storm clustering method is proposed that can automatically group individual storm events into a limited set of spatial clusters.*

## 1. Introduction

The advances in real-time observation, information technology and modeling enable the paradigm shift of short-term weather forecast from static model forecast to dynamic and adaptive model forecast. A traditional forecast runs a mesoscale model at fixed time interval over a region of interest. As a result, it cannot respond well to the upcoming weather events, which are relatively short lived and change over time. On the contrary, a dynamic forecast is triggered by weather events and model run focuses on the regions of interest where weather events are most active. If the real-time observations are assimilated in the model run, a forecast becomes adaptive. Dynamic and adaptive weather forecast can respond to weather events more accurately and timely than static model forecast.

[1] Corresponding author address: University of Alabama in Huntsville, 301 Sparkman Dr. Huntsville, AL 35899. E-mail: xli@itsc.uah.edu.

The Linked Environments for Atmospheric Discovery (LEAD) project (Droegemeier et al., 2004) is a large-scale, interdisciplinary NSF-funded research project that aims to address fundamental information technology and meteorology research challenges in dynamic and adaptive mesoscale weather forecast. The objective of the project is to develop a LEAD cyber-infrastructure (LEAD-CI) for identifying, accessing, decoding, assimilating, analyzing, mining, and visualizing a broad array of meteorological data and model output necessary for next-generation weather forecast (Droegemeier et al., 2005). The LEAD-CI takes a service-oriented architecture (SOA) and provides tools and services that allow users to automatically spawn weather forecast models in response to real time weather events. One of the integral components for dynamic weather forecast is the automatic weather event detection from real-time observational data. In LEAD-CI, this is accomplished through the Calder stream processing service, which allows users SQL query access to collections of live data streams. Figure 1 shows schematic diagram of the Calder stream processing service. Calder takes requests in the form of "User query", normally request from other services, filters the input data streams that users are interested, and dispatches the data to the event detection module for storm event detections. Vijayakumar et al (2006) presented in detail the framework of Calder stream processing service. In Vijayakumar et al (2006), the event detection module was the Mesocyclone Detection Algorithm (MDA), which detected tornadic events using the WSR88D radar velocity data. Event trigger was dispatched from the service in response to each individual storm detected from radar observation. During a stormy day, tens to hundreds of events could be detected over a region of interest.

Even with the grid computational resources, it is inconceivable to allow hundreds of models running simultaneously on the LEAD in response to all the individual storms detected. It may also

not be necessary. More often the storm events detected reside close to each other and can be clustered into groups. Therefore, instead of responding to individual storms detected, it will be more appropriate for the LEAD system to respond to the clusters of storms. In this way, the event detection service requires an event detection algorithm followed by a storm clustering algorithm. The objective of this research is to investigate clustering algorithms that can effectively and automatically group the storm events into spatial clusters. Two clustering algorithms, the *k*-means algorithm and the DBSCAN algorithm, are investigated for their storm clustering performance. Determining the optimal number of clusters in a data set is a common challenge for clustering applications. In this study, we examine two statistical indices for their capabilities in determining optimal number of clusters.

The reminder of the paper is organized as follow. Section 2 introduces the *k*-means and the DBSCAN clustering algorithms. This section also presents the two statistical indices used for estimating the optimal number of clusters in a data set. Section 3 presents the performance comparisons of the clustering algorithms as well as the statistical indices. Our conclusions are summarized in section 4.

## 2. Storm events clustering

### 2.1 Clustering algorithms

A storm event detected from a storm detection algorithm such as the MDA is normally characterized by its geospatial attributes, normally the latitude and longitude and its physical attributes, such as size, base height, depth, and radar reflectivity. Therefore, classical clustering methods can be directly applied to storm event data.

In this study, we examined two clustering algorithms: *k*-means algorithm (MacQueen, 1967) and DBSCAN algorithm (Ester et al. 1996). The *k*-means algorithm is one of the simplest and most popular algorithms for clustering analysis. It is a partitioning-based clustering algorithm. Given a number of clusters in a data set, the *k*-means algorithm iteratively determines the center for each cluster and assigns each data sample to a cluster to whose center the data sample is closest. By representing each sample with its cluster center, an overall error, normally the root-mean-square error (RMS), is calculated at each iteration. This iterative clustering process terminates when the overall error reaches minimum value. Since our focus for spatial clustering is to group storm events into local regions of interest, latitude and longitude are the only features used in the *k*-means clustering. One of the primary challenges in using *k*-means algorithm is the selection of the number of clusters for a data set.

The DBSCAN algorithm is a density-based clustering algorithm, which regards a spatial cluster as a region in data space that contains data samples with certain density. Data density around a data sample is determined as the number of samples *N* within a distance $\varepsilon$ to the sample. Two data samples are connected if their distance is less than $\varepsilon$. DBSCAN algorithm connects neighboring samples into spatial clusters. Unlike the *k*-means algorithm that performs well for spherical-shaped clusters, the DBSCAN algorithm can identify a spatial cluster of any shape. The *N* and $\varepsilon$ parameters in the DBSCAN algorithm determine the number of clusters in a data set and the clustering performance.

### 2.2 Number of clusters

Clustering algorithms such as the *k*-means algorithm explicitly require number of clusters as an input parameter. The *N* and $\varepsilon$ parameters in the DBSCAN algorithm implicitly indicate the number of clusters in a data set. One common approach to automatically determine the optimal number of cluster is to apply some criterion to evaluate the fitness between a data set and clustering result where the optimal number of clusters produces best fit. In this study, we examine two of such kind of statistical criteria: *Hargitan index* (1975), a statistical index to examine the relative change of fitness as number of clusters changes, and *average silhouette (AS) index.*

Assume a data set containing *N* samples, $\boldsymbol{X}_i$, $i=1,N$ where $\boldsymbol{X}_i$ is a $\boldsymbol{M}$-component vector, representing $\boldsymbol{M}$ features for sample *i*. For a clustering result with *k* clusters, the overall fitness for the clustering can be expressed as the square of error for all samples:

$$err(k) = \sum_{i=1}^{k} \sum_{j=1,\, j \in Ci}^{N} d^2(Xj, X_{ci}),$$

Where *d* is the distance between data sample $\boldsymbol{X_j}$ and the center $\boldsymbol{Xc_i}$ that it belongs to. *err(k)* is the total square of error for *k* cluster partitioning.

Then, *Hartigan* index *H(k),* for *k* partitioning, is expressed as follow:

$$H(k) = (n-k-1)\frac{err(k) - err(k+1)}{err(k+1)}$$

Since *err(k)* is monotonically non-increased with increasing *k*, the ratio is a relative measure of the reduction of square error when number of clusters increases from *k* to *k*+1. The multiplier correction term of *(N-k-1)* is a penalty factor for large number of cluster partitioning. The optimal *k* number is the one that maximizes the *H(k).*

*AS* index is another statistical measure that is often used for determining number of clusters. Still, assume a data set as given above. The average distance of a data sample $\boldsymbol{X_i}$ to all data samples $\boldsymbol{X_j}$ that belongs to cluster *C* is as follow:

$$d(i,c) = \frac{1}{N_c}\sum_{j\in C} d(X_i, X_j),$$

where $N_c$ is the total number of data samples in cluster *C*. Then, the average intra-cluster distance *a(i)* for $\boldsymbol{X_i}$ is

$$a(i) = d(i, C_i), i \in C_i,$$

where $C_i$ is the cluster that *i* belongs to. The minimum average inter-cluster distance *b(i)* for $\boldsymbol{X_i}$ is

$$b(i) = \min d(i, C_j), i \notin C_j, C_j \neq C_i,$$

where $C_j$ is the cluster that *i* does not belongs to.

The silhouette *Sil(i)* for sample *i* is then defined as

$$Sil(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

The average silhouette for *k* partitioning, *AvgSil(k),* is then defined as

$$AvgSil(k) = \frac{1}{N}\sum_{i=1}^{N} Sil(i)$$

The optimal *k* number is the one that maximizes the *AvgSil(k).*

## 3. Experimental Results and Discussions

### 3.1 Radar data and storm event detection

In this study, the storm events were detected from the WSR88D radar measurement taken from the 134 sites covering the CONUS. The data set contains radar measurement from 1:00pm to 6:00pm EST on March 29, 2007. A threshold-based storm detection algorithm (SDA) was applied to all the radar volume scans collected during this time period for storm events. The SDA algorithm detected radar data volumes with reflectivity values larger than 40 DBz and then groups these data volumes into 3D regions of interest using region-growing technique. Each of these regions of interest was considered as a single storm event. There were total of 4258 storm events detected during the five hours time period. Since one radar volume scan takes about 5 minutes and the radar measurements are not synchronized over the CONUS, for the clustering purpose; we divided each hour into 4 time intervals with each one having 15 minutes. Thus, there were total of 20 time intervals during which storm events detected were clustered.

### 3.2 *k*-means algorithm vs. DBSCAN algorithm

We compared the performances of the two different clustering algorithms. Figure 2 shows the storm events detected during the time period from 1:00 pm to 1:15 pm. Figures 2a and 2c shows the *k*-means clustering results with *k*=2 and 3, respectively. Figures 2b and 2d shows the clustering results using the DBSCAN algorithms. In DBSCAN algorithm, the *N* is chosen as 3, and $\varepsilon$ values are 9.0 and 6.3 for Figure 2b and Figure 2d, respectively. For comparison purpose, these $\varepsilon$ values are selected so that the number of clusters in Figures 2b and 2d are 2 and 3, respectively. Based on visual inspection, the data set appears to have 3 spatial clusters: C1 is on the upper right corner, C2 is on the lower right part of the figure, and C3 in the upper left part of the figure, though the storms in C3 are less dense than in clusters C1 and C2. When the number of clusters is restricted to 2, the *k*-means algorithm merges clusters C2 and C3, while the DBSCAN algorithm merges clusters C1 and C2, as Figures 2a and 2b show. It seems more reasonable to merge C1 and C2 instead of C2 and C3. When the number of cluster is set to 3, the results from the DBSCAN algorithm agree with our visual inspection; while the *k*-means algorithm still merges clusters C2 and C3, it splits the cluster C1 into 2 parts. *These initial results indicate that the DBSCAN algorithm performs better storm clustering than that of the k-means algorithm.*

### 3.3 Number of clusters

We also compared the performances of *Hartigan* index and *AS* index in determining optimal number of clusters in a data set. Figure 3a shows the *Hartigan* index as function of number of cluster k, and Figure 3b shows the *AS* index as

function of number of cluster k for time period between 1:00pm and 1:15pm. The optimal number of cluster using *Hartigan* index is 3. From Figure 3a, the value of *Hartigan* index at $k = 3$ is significantly larger than the values for rest of the *k*. In other word, *Hartigan* index strongly suggests that the there are three clusters in the data set. The optimal *k* value from *AS* index is 5, as shown in Figure 3b. However, the values of *AS* index change slightly as *k* value ranges from 2 to 5. Visual inspection of Figure 2 suggests that storm events can be properly clustered into three groups. The optimal *k* value from *Hartigan* index agrees with the visual inspection.

Further examination of all the storm data set shows that the two indices indicate the same cluster number 14 out of the 20 data sets. For the 6 remaining data sets, cluster numbers differ by 1 for 3 data sets. They differ by 2 for one data set and differ by 3 for another data set. The largest discrepancy occurs at time period 13:30 – 13:45 pm where they differ by 15. Results from *Hartigan* index agree well with our visual inspection. *This implies that Hartigan index provides better and more reliable performance than that of AS index.*

## 4 Conclusions

In this study, we investigated clustering methods to automatically group the individual storm events into local clusters. Two clustering algorithms, the *k*-means algorithm and the DBSCAN algorithm, were examined and their clustering performances were compared. It was found that the DBSCAN algorithm has a superior performance than that of the *k*-means algorithm. Also, we compared two statistical indices for determining the number of clusters in a storm event data set. It was found that for most of the data sets, the optimal number of clusters determined using the *Hartigan* index and the *AS* index were the same. However, the clustering performance using the *AS* index was more sensitive to the storm event distributions. As a result, we propose using the combination of DBSCAN algorithm with *Hartigan* index measure for automatic storm event clustering.

## Reference


Droegemeier, K. K. and Co-Authors, 2004: Linked environments for atmospheric discovery (LEAD): A cyberinfrastructure for mesoscale meteorology research and education, *20th Conf. on Interactive Info. Processing Systems for Meteorology, Oceanography, and Hydrology*, Seattle , WA , Amer. Meteor. Soc.

Droegemeier, K. K., D. Gannon, D. Reed, B. Plale, J. Alameda, T. Baltzer, K. Brewster, R. Clark, B. Domenico, S. Graves, E. Joseph, V. Morris, D. Murray, R. Ramachandran, M. Ramamurthy, L. Ramakrishnan, J. Rushing, D. Weber, R. Wilhelmson, A. Wilson, M. Xue, and S. Yalda, 2005: Service-Oriented Environments in Research and Education for Dynamically Interacting with Mesoscale Weather. *IEEE Computing in Science & Engineering*, 7, 24-32.

Ester, M., H.-P. Kriegel, J. Sander, and X. Xu, 1996: A density-based algorithm for discovering clusters in large spatial databases with noise. In proceedings of *2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD-96)*, Portland, OR, 226-231.

Hartigan, J. A., 1975: Clustering Algorithms. *Wiley Series in Probability and Mathematical Statistics*, John Wiley & Sons, Inc.

MacQueen, B., 1967: "Some Methods for classification and Analysis of Multivariate Observations, *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability"*, Berkeley, University of California Press, 1:281-297

Vijayakumar, N., B. Plate, R. Ramachandran, and X. Li, 2006: Dynamic filtering and mining triggers in mesoscale meteorology forecasting, *IEEE International Symposium on Geoscience and Remote Sensing*, July 31- Aug. 4, 2006, 2449-2452.

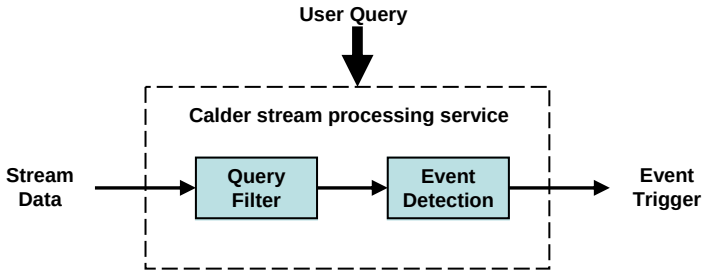


Figure 1 Schematic diagram for Calder stream processing service

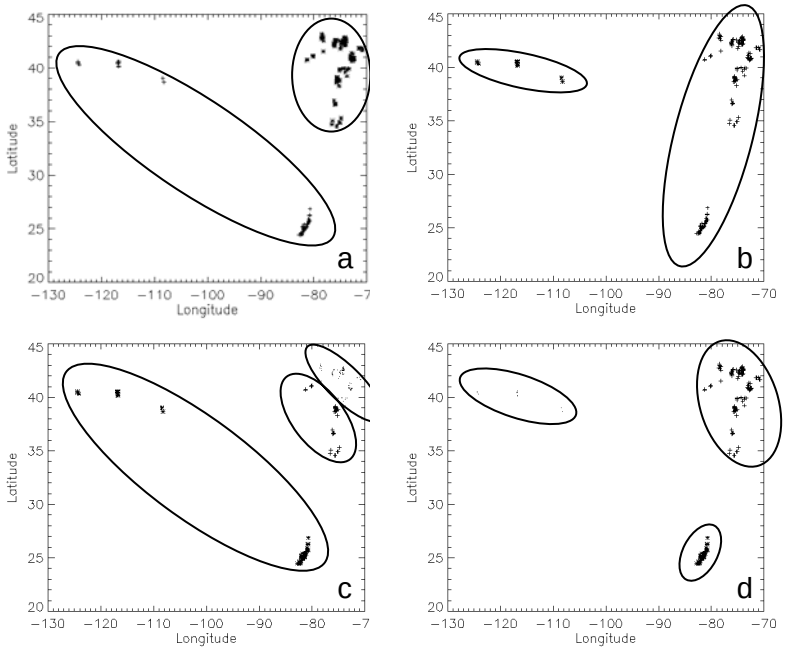


Figure 2 clustering results using k-means and DBSCAN algorithms for time period 1:00pm – 1:15pm. (a) k-means algorithm with k = 2, (b) DBSCAN algorithm with k = 2, (3) k-means algorithm with k = 3, and (d) DBSCAN algorithm with k = 3

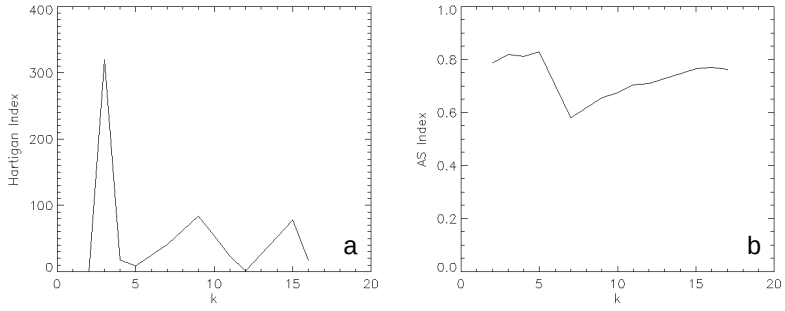


Figure 3 a) the Hartigan index and b) the AS index, as function of number of cluster k for time period 1:00pm – 1:15pm.