University of Nebraska - Lincoln

DigitalCommons@University of Nebraska - Lincoln

CSE Conference and Workshop Papers

Computer Science and Engineering, Department
of

2009

Density-Based Clustering of Polygons

Deepti Joshi

University of Nebraska-Lincoln, djoshi@cse.unl.edu

Ashok Samal

University of Nebraska-Lincoln, asamal1@unl.edu

Leen-Kiat Soh

University of Nebraska-Lincoln, lsch2@unl.edu

Follow this and additional works at: <https://digitalcommons.unl.edu/cseconfwork>

Part of the [Computer Sciences Commons](#)

Joshi, Deepti; Samal, Ashok; and Soh, Leen-Kiat, "Density-Based Clustering of Polygons" (2009). *CSE Conference and Workshop Papers*. 26.

<https://digitalcommons.unl.edu/cseconfwork/26>

This Article is brought to you for free and open access by the Computer Science and Engineering, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in CSE Conference and Workshop Papers by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

Density-Based Clustering of Polygons

Deepti Joshi, Ashok K. Samal, *Member, IEEE* and Leen-Kiat Soh, *Member, IEEE*

Abstract – Clustering is an important task in spatial data mining and spatial analysis. We propose a clustering algorithm P-DBSCAN to cluster polygons in space. P-DBSCAN is based on the well established density-based clustering algorithm DBSCAN. In order to cluster polygons, we incorporate their topological and spatial properties in the process of clustering by using a distance function customized for the polygon space. The objective of our clustering algorithm is to produce spatially compact clusters. We measure the compactness of the clusters produced using P-DBSCAN and compare it with the clusters formed using DBSCAN, using the Schwartzberg Index. We measure the effectiveness and robustness of our algorithm using a synthetic dataset and two real datasets. Results show that the clusters produced using P-DBSCAN have a lower compactness index (hence more compact) than DBSCAN.

I. INTRODUCTION

Clustering is the process of unsupervised classification that is fundamental to spatial data mining and spatial analysis. Several spatial clustering algorithms have been proposed in the past (see Section 2.1). However, most of them are focused on clustering point data sets. There are several applications of spatial clustering where clustering algorithms for point datasets may not give efficient results. This mainly happens when polygons need to be clustered instead of points. For example, an important application of polygonal clustering is the process of regionalization. Regionalization is the process of region building where smaller units (polygons) are grouped together into larger contiguous regions based on some attribute or criteria. Thus, regionalization produces clusters of polygons that are spatially compact and contiguous. If polygons are indeed represented as points and clustering is performed, the spatial information and relationships between polygons are not captured and utilized during the clustering process. Due to the inadequacies of the point-based clustering algorithms

new clustering algorithms need to be developed in order to cluster polygons. In this paper we propose a novel algorithm P-DBSCAN for clustering polygonal datasets.

Our algorithm P-DBSCAN is based on the well established density-based clustering algorithm DBSCAN [7]. There are several advantages of using DBSCAN as our reference algorithm. First, it has the ability to discover clusters of arbitrary shapes such as linear, concave, and oval. Second, DBSCAN does not require the number of clusters to be determined in advance. Finally, DBSCAN is scalable to be used with large databases. The new algorithm P-DBSCAN extends DBSCAN to cluster *polygons* instead of points by redefining the concepts of the neighborhood of a polygon, core polygon, border polygon, and noise polygon. The clustering is done based on the distance between two polygons leading to the polygons close to each other being clustered together, and thus resulting in spatially compact clusters. Note that a key component of our P-DBSCAN algorithm is the calculation of the distance function (see Section 3.2). Using this distance function, both contiguous polygons and disjoint polygons can be clustered using our novel algorithm. When the polygons are contiguous in space, the extent of the boundary shared by two polygons is taken into account while computing the distance between them. On the other hand, if the polygons are disjoint, the shared boundary component is ignored. PDBSCAN is not restricted to polygons in 2-D space only, and is applicable to polygons in n -dimensional space, with $n > 2$.

The rest of the paper is organized as follows. Section 2 presents the related work giving a background on spatial clustering and density-based spatial clustering. Section 3 defines the density-based concepts for polygons, our methodology for computing the distance between two polygons, and explains our algorithm in detail. Section 4 presents an application of our clustering algorithm. Finally, our conclusion and directions for future work are given in Section 5.

II. RELATED WORK

A. Spatial Clustering Algorithms

Clustering algorithms can be categorized into five main types: Partitional, Hierarchical, Density-based, Grid-based, and Model-based clustering algorithms. In Partitional algorithms, partitions of a database D are

This material is based upon work supported by the National Science Foundation under Grants No. 0219970 and 0535255.

Deepti Joshi is with the Computer Science and Engineering Department, University of Nebraska-Lincoln, NE 68588, USA (e-mail: djoshi@cse.unl.edu).

Ashok K. Samal is with the Computer Science and Engineering Department, University of Nebraska-Lincoln, NE 68588, USA (e-mail: samal@cse.unl.edu).

Leen-Kiat Soh is with the Computer Science and Engineering Department, University of Nebraska-Lincoln, NE 68588, USA (e-mail: lksoh@cse.unl.edu).

developed, and a set of clusters are formed. The number of clusters generated has to be specified in advance. The cluster similarity is measured with respect to the mean value (cluster center) of the objects in a cluster. Examples are PAM [13], CLARA [13], and CLARANS [13].

Hierarchical algorithms create a hierarchical decomposition of the database. This hierarchical decomposition is represented as a dendrogram. Each level of the dendrogram represents a set of clusters. Thus, a set of nested clusters organized as a hierarchical tree are produced. As a result the initial knowledge of the number of clusters is no longer required. However, a termination condition needs to be specified. Examples of hierarchical clustering are CURE [14] and BIRCH [19].

Density-based clustering algorithms are based on the idea that objects which form a dense region should be grouped together into one cluster. These algorithms search for regions of high density in a feature space that are separated by regions of lower density. Thus, density-based methods can be used to filter out noise, and discover clusters of arbitrary shape. Examples of density-based clustering algorithms are DBSCAN [7], DENCLUE [10], and OPTICS [11].

Grid-based algorithms are based on multiple level grid structure. The entire space is quantized into a finite number of cells on which operations for clustering are performed. Summarized information about the area covered by each cell is stored as an attribute of the cell. The main advantage of this approach is its fast processing time. However, the summarized information leads to loss of information. Examples of grid-based clustering algorithms are STING [17], WaveCluster [16], and CLIQUE [1].

In model-based algorithms a model is hypothesized for each of the clusters and the idea is to find the best fit of that model to each cluster. They are often based on the assumption that the data are generated by a mixture of underlying probability distributions. COB-WEB [2] is an example of this approach.

We select the density-based approach for clustering polygons since there is no need to know the number of clusters in advance as required in partitioning algorithms, nor is there a need to store summarized information as in grid-based algorithms. Moreover, polygons in geographic space and in many other domains naturally respond to the density-based approach. For example, in geographic space, we have a set of contiguous polygons, and another set of polygons located far away from the first set. At a larger scale, these two sets will belong to a cluster each, thus corresponding to clusters formed where the object density is high.

B. Density-Based Concepts for Points

A density-based clustering algorithm hinges upon the assumption that a valid cluster must have sufficient density. Ester et al. proposed a density-based clustering algorithm used for clustering point datasets, called DBSCAN [7]. Here we list the main concepts of density for points as defined in [7]. These concepts are later (see Section 3.1) extended in our clustering algorithm P-DBSCAN for clustering polygons.

Definition 1: (ϵ -neighborhood of a point) The ϵ -neighborhood of a point p , denoted by $N_\epsilon(p)$, is defined by $N_\epsilon(p) = \{q \in D \mid \text{dist}(p, q) \leq \epsilon\}$.

Definition 2: (directly density-reachable) A point pp is *directly density-reachable* from a point qq wrt. ϵ , $MinPts$ if 1) $p \in N_\epsilon(q)$ and 2) $|N_\epsilon(q)| \geq MinPts$ (core point condition).

Directly density-reachable is symmetric for pairs of core points. In general, however, it is not symmetric if one core point and one border point are involved.

Definition 3: (density-reachable) A point p is *density-reachable* from a point q wrt. ϵ , $MinPts$ if there is a chain of points $p_1, \dots, p_n, p_1 = q, p_2 = p$ such that p_{i+1} is directly density-reachable from p_i .

Definition 4: (density-connected) A point pp is *density-connected* to a point qq wrt. ϵ , and if there is a point o such that both, p and q are density-reachable from o wrt. ϵ , $MinPts$. Density-connectivity is a symmetric relation. For density-reachable points, the relation of density-connectivity is also reflexive.

Definition 5: (cluster) Let D be a database of points. A *cluster* C wrt. ϵ , $MinPts$ is a non-empty subset of D satisfying the following conditions:

- 1) $\forall p, q$: if $p \in C$ and q is density-reachable from p wrt. ϵ and $MinPts$, then $q \in C$. (Maximality)
- 2) $\forall p, q \in C$: p is density-connected to q wrt. ϵ and $MinPts$. (Connectivity)

Definition 6: (noise) Let $C_1, \dots, C_k$ be the clusters of the database D wrt. parameters ϵ and $MinPts$, then we define the *noise* as the set of points in the database D not belonging to any cluster C_i , i.e. $noise = \{p \in D \mid \forall i: p \notin C_i\}$.

III. P-DBSCAN ALGORITHM

A. Density-Based Concepts for Polygons

Since polygons are spread out in space, factors that would have no effect on points—such as topology and direction—come into play. Also, if the polygons are share boundaries, then two polygons sharing a larger extent of their boundary should be considered closer to

each other as compared to two polygons sharing a very small portion of their boundaries. This conclusion follows from the observation that more close two polygons are to each other, more similar they will be in their characteristics. As a result of these factors, some of the density-based concepts for points do not directly apply to polygons. Mainly, the concept of a *core polygon* and its neighborhood are fundamentally different from that of a core point. Once a core polygon is defined, and the polygons that belong to its neighborhood, the same concepts of directly-density reachable, density-reachable, and density-connected for points can then be applied to polygons. In the following, we formalize the density-based concepts for polygons.

ϵ -neighborhood of a Polygon: The ϵ -neighborhood of a polygon p , denoted by $N_\epsilon(p)$, is defined by $N_\epsilon(p) = \{q \in D \mid \text{dist}(p, q) \leq \epsilon\}$, where D is the data set of polygons, and $\text{dist}(p, q)$ is defined as the distance between polygons p and q . For example in Figure 1, the ϵ -neighborhood of the polygon p is $N_\epsilon(p) = \{a, b, c, d, e, f, g\}$.

Radial Spatial Neighborhood of a Polygon: The neighborhood of a polygon can be further partitioned. That is, $N_\epsilon(p) = \bigcup_{i=1}^R N_{\epsilon,i}(p)$ such that R is the number of equal-size sectors radially partitioning the space around the polygon p . The definition of $N_{\epsilon,i}(p)$ extends directly from the ϵ -neighborhood of the polygon, but only looks at the sector indexed by i . Figure 1 shows an example of the radial spatial neighborhood of a polygon p (shaded).

Fig. 1. $R = 8$ Radial spatial partitions of a polygon's neighborhood. Note that here the first sector is S_1 as shown, and the ordering is clockwise. This is arbitrary for illustration purpose.

The radial spatial neighborhood of polygon p in Figure 1 is divided into 8 sectors: $S_1, \dots, S_8$. As shown in Figure 1, $N_{\epsilon,1}(p) = \{b, c\}$, $N_{\epsilon,2}(p) = \{c, d\}$, $N_{\epsilon,3}(p) = \{d, e\}$, $N_{\epsilon,4}(p) = \{e, f\}$, $N_{\epsilon,5}(p) = \{f, g\}$, $N_{\epsilon,6}(p) = \{g\}$, $N_{\epsilon,7}(p) = \emptyset$, $N_{\epsilon,8}(p) = \{8\}$. Thus, $\bigcup_{i=1}^R N_{\epsilon,i}(p) = \{a, b, c, d, e, f, g\}$ which is the same as $N_\epsilon(p)$.

Core Polygon: A *core polygon* c is defined as a polygon that has at least a minimum number of polygons ($MinPolys$) within its ϵ -neighborhood, and there are at least a minimum number of radial spatial partitions ($MinS$) that are non-empty, i.e. $[Count_{i=1}^R (N_{\epsilon,i}(c) \neq \emptyset)] \geq MinS$. For example, in Figure 2, if $\epsilon = 1$, $MinPolys = 4$ and $MinS = 8$, p , o and q are core polygons.

Border Polygon: A *border polygon* b is defined as a polygon that has more than $R - MinS$ of its radial spatial partitions empty, i.e. $[Count_{i=1}^R (N_{\epsilon,i}(b) = \emptyset)] > R - MinS$, where R is the total number of partitions. For example, in Figure 2 with $\epsilon = 1$, $MinPolys = 4$ and $R = 8$, and $MinS = 8$, b is a border polygon, since $R - MinS = 0$.

Outlier Polygon: An *outlier polygon* is defined as a polygon that does not have any polygons within the threshold distance of ϵ .

Directly Density-Reachable: A polygon p is *directly density-reachable* from a polygon q wrt ϵ , if

- 1) $p \in N_\epsilon(q)$ and
- 2) q is a core polygon.

Directly density-reachable is symmetric for pairs of core polygons. In general, however, it is not symmetric if one core polygon and one border polygon are involved. For example, in Figure 2 polygon a is *directly density-reachable* from a polygon p , however polygon p is not *directly density-reachable* from a polygon a .

Density-Reachable: A polygon p is *density-reachable* from a polygon q if there is a chain of polygons $p_1, \dots, p_n$ where $p_1 = q$ and $p_n = p$ such that p_{i+1} is directly density-reachable from p_i where $\{i = 1 \text{ to } n - 1\}$. In Figure 2 polygons p is density-reachable from polygon q .

Fig. 2. Synthetic set of polygons (Red – Core Polygon, Green – ϵ -neighborhood of the core polygons)

Density-Connected: A polygon p is *density connected* to a polygon q if there is a polygon o such that both, p and q are density-reachable from o . In Figure 2, polygon a and polygon b are density-reachable from polygon o , and thus are density-connected to each other.

Cluster: A cluster C wrt. ε is a non-empty subset of D satisfying the following conditions:

- 1) *Maximality:* $\forall p, q | p \in C, q \in D$ q is density-reachable from p , then $q \in C$.
- 2) *Connectivity:* $\forall p, q \in C: p$ is density-connected to q .

B. Distance Function for Polygons

Each polygon is represented as a set of vertices that form the boundary of the polygon. We use the Hausdorff distance as the basis for computing the distance between two polygons in the boundary space. The *Hausdorff distance* between two sets of points [Rote 1991] is defined as the maximum distance of points in one set to the nearest point in the other set. Formally, the Hausdorff distance (D_h) from set A to set B is defined as

$$D_h(A, B) = \max_{a \in A} (\min_{b \in B} d(a, b)) \quad (1)$$

where a and b are points of sets A and B , respectively, and $d(a, b)$ is any distance metric between the two points a and b . The distance metric used within Hausdorff distance in order to calculate the distance between two points is the Euclidian distance.

If the boundaries of the polygons p and q are represented by two sets of points A and B respectively, we use the following defined distance measure (d_h) between two polygons

$$d_h(p, q) = \max (D_h(A, B), D_h(B, A)) \quad (2)$$

Intuitively, we expect the distance between two polygons with shared boundary to be less. However, the standard Hausdorff distance is defined on the set of points and does not incorporate any sharing of the boundary. In order to incorporate this, we define a new distance measure, called the *boundary adjusted Hausdorff distance*, that is inversely proportional to the length of the shared boundary between the two polygons, between two polygons p and q as follows:

$$d_{hs}(p, q) = \left(1 - \frac{2S_{pq}}{S_p + S_q}\right) \times d_h(p, q) \quad (3)$$

where d_h is the original standard Hausdorff distance, S_p and S_q are the perimeter lengths of polygons p and q , respectively, and S_{pq} is the length of their shared boundary. This distance, d_{hs} , is smaller than the standard Hausdorff distance when two polygons have shared boundary, and becomes the standard Hausdorff distance when two polygons have no shared boundary, i.e., when $S_{pq} = 0$. We use twice the shared distance in the definition to balance the effect of the denominator.

C. P-DBSCAN Algorithm

Our algorithm works similar to DBSCAN where we select a polygon p from the dataset D and check if it has been assigned to a cluster already. If the polygon is still unclassified, then the *ExpandCluster* routine is called. As in DBSCAN, *ExpandCluster* is the where the cluster assignment is done. P-DBSCAN checks whether a polygon is a core polygon or not by calling the *Expandable* method. This method generalizes the method of checking for the coreness of a polygon or any other object being clustered, as opposed to DBSCAN that implicitly checks only for the MinPts condition. If a polygon is classified as a core polygon, its neighbors are retrieved from the database and assigned to the same cluster as the core polygon. Given below is our proposed P-DBSCAN Algorithm.

P-DBSCAN

Input: $D, \varepsilon, MinPolys$

Output: Set of Clusters

1. Initially all polygons are UNCLASSIFIED
2. *ClusterId* is initialized
3. For each polygon p in D
 - 3.1. **If** its *ClusterId* is UNCLASSIFIED **then** call *ExpandCluster*.
 - 3.1.1. **If** *ExpandCluster* returns True **then** increment *ClusterId*

Expandable

Input: p

Output: True or False

1. **If** p is surrounded by polygons in at least *MinS* radial spatial partitions **then**
 - 1.1. Get the ε -Neighborhood of p .
 - 1.2. **If** ε -Neighborhood of p contains *MinPolys* polygons **then**
 - 1.2.1. Return True
2. **Else** return False.

ExpandCluster

Input: $p, ClusterId$

Output: True or False

1. **If** p is *Expandable* **then**
 - 1.1. Set the ClusterID of p to *ClusterId*
 - 1.2. For each neighbor of p , call the *ExpandCluster* routine.
 - 1.2.1. Return True.
2. **Else** return False.

DBSCAN now becomes a special case of P-DBSCAN. The time complexity of our algorithm remains the same as DBSCAN that is $n \log(n)$ where n is the size of the database.

IV. RESULTS

To show the effectiveness of our algorithm we have conducted several experiments and compared our results with DBSCAN. The input to the P-DBSCAN algorithm are the polygons, a pre-defined ϵ and a pre-defined $MinPolys$. R is set to 4, and $MinS$ is set to 4 for all experiments as well. The input to the DBSCAN algorithm are the centroids of the polygons, a pre-defined ϵ and a pre-defined $MinPts$. To demonstrate the robustness of our algorithm we use two different experiments. We first use a synthetic dataset which is a 10×10 grid of 1×1 unit squares. We then use two real datasets from a practical application, i.e. the census tracts of two states in USA – Nebraska and South Dakota. When DBSCAN was applied on these datasets, the Euclidean distance was computed between the centroids of the polygons in order to measure how close they are to each other. P-DBSCAN uses the modified Hausdorff distance function as described in Section 3.2. All the three datasets are sets of contiguous polygons. Thus, both the algorithms DBSCAN and P-DBSCAN when applied with the appropriate input parameters should result in a single cluster consisting of all the polygons. The details of the experiments are as follows:

A. Experiment on Synthetic Dataset

The first set of experiments were conducted using a 10×10 grid resulting in a dataset with 100 polygons all of the same size and shape. The reason to use this dataset was to show that P-DBSCAN produces the same results as DBSCAN when all the polygons are equidistant from each other, making DBSCAN a special case of P-DBSCAN. In the first test, we applied $\epsilon = 0.5$ which resulted in zero clusters (Figure 3) since the distance was too small to include any other polygon in its neighborhood. When $\epsilon = 1.5$ (Figure 4), all the polygons were grouped together in the same cluster by both the algorithms, i.e. DBSCAN and P-DBSCAN.

Fig. 3. Result of clustering using DBSCAN (a) Polygons used for clustering (b) Expanded version of dataset showing $\epsilon = 0.5$

Figure 4 shows how the cluster grows upon the application of the DBSCAN algorithm to the dataset. Figure 4(b) shows the first core polygon in red. The surrounding polygons shown in green belong to the ϵ -neighborhood of the core polygon. Figure 4(c) shows the next core polygon detected. Finally Figure 4(e) shows the entire cluster. All the polygons except the

four corner polygons shown in green were marked as core polygons by the algorithm.

Fig. 4. Result of clustering using DBSCAN (a) $\epsilon = 1.5, MinPts = 5$ (b) First core polygon (Red) and its ϵ -neighborhood (Green) (c) Consecutive core polygon detected and its ϵ -neighborhood (d) Further progression of core polygon detection belonging to the same cluster (e) Final result – All polygons belong to the same cluster.

We examine the performance of P-DBSCAN using the same dataset. The spatial neighborhood of a core polygon is divided into $R = 4$ radial partitions with $MinS = 4$, and $MinPolys = 5$. The result of clustering the polygons shown in Figure 4(a) can be seen in Figure 5.

Fig. 5: Result of clustering using P-DBSCAN (a) Polygons used for clustering $\epsilon = 1.5, MinPolys = 5, MinS = 4$ (b) First core polygon (Red) and its ϵ -neighborhood (Green) (c) Further progression of core polygon detection belonging to the same cluster (d) Final result – All polygons belong to the same cluster.

We can see in the above figures that while the core points and core polygons are not the same, both the algorithms resulted in the same cluster consisting of all the polygons in the grid.

B. Experiments on Real Datasets

Experiments were conducted on two sets of real data – the Nebraska census tract dataset, and South Dakota census tract dataset. The Nebraska dataset (Figure 6) consists of a set of 505 contiguous polygons. Both the algorithms DBSCAN and P-DBSCAN were applied to this dataset using different values of ϵ , $MinPts$, and $MinPolys$.

The results for DBSCAN with different values of ϵ and $MinPts$ can be seen in Figure 7. We start with the value ϵ as average distance between the centroids of the polygons in the dataset which is 0.75, and $MinPts = 2$ (Figure 7(a)). We find that all the polygons are clustered together to form one large cluster. When

Fig. 6. Census Tract Polygons in Nebraska dataset

$MinPts$ was increased to 5 (Figure 7(b)), the number of clusters did not increase and some polygons were left out from the cluster. With a smaller value of ϵ more clusters are produced with a large number of polygons being left unclustered.

Fig. 7. Results of clustering using DBSCAN (a) $\epsilon = 0.75, MinPts = 2$ (b) $\epsilon = 0.75, MinPts = 5$ (c) $\epsilon = 0.25, MinPts = 5$ (d) $\epsilon = 0.5, MinPts = 5$

The results for P-DBSCAN with different ϵ value and $MinPolys$ can be seen in Figure 8. Here too we start with $\epsilon = 0.75$. $MinPolys$ was set to 1, 2, and 5. With $\epsilon = 0.75, MinPolys = 1$ (Figure 8(a)), it was seen that all the polygons belonged to a cluster leaving no polygons unclustered. As $MinPolys$ was increased (Figure 8(b) & 8(c)), the number of polygons left unclustered increased. When ϵ was increased, number of polygons belonging to a cluster reduced even further, leaving a lot white space or unclustered polygons within the dataset (Figure 8(d)). On the other hand, when ϵ was increased, and reached to a value of 2, all the polygons were clustered together to belong to the same cluster. The same trend of number of clusters detected with increasing ϵ was seen here as well, with all the polygons belonging to the same cluster when $\epsilon = 2.0$ (Figure 8(f)).

In order to compare the results of both the algorithms shown above, we compute the *compactness of a cluster* using the *Schwartzberg Index*[17]. It measures the compactness of a cluster as the square of the perimeter of the cluster divided by the area of the cluster. The lower the value of this index, the more compact the cluster is.

Fig. 8. Results of clustering using P-DBSCAN (a) $\epsilon = 0.75, MinPolys = 1$ (b) $\epsilon = 0.75, MinPolys = 2$ (c) $\epsilon = 0.75, MinPolys = 5$ (d) $\epsilon = 0.5, MinPolys = 5$ (e) $\epsilon = 1.0, MinPolys = 5$ (f) $\epsilon = 2.0, MinPolys = 5$

The compactness index was computed for the clusters formed by DBSCAN and P-DBSCAN for the Nebraska dataset. In order to compare the compactness of the clusters formed by both the algorithms, we computed the average all the clusters formed at a given ϵ and $MinPolys = 5$. The results are shown in Figure 9. The number above each bar represents number of clusters.

Fig. 9. Compactness Ratio for clusters formed using DBSCAN and P-DBSCAN

As shown in Figure 9, P-DBSCAN produces clusters with a lower compactness index. This implies that the clusters formed using P-DBSCAN are spatially more compact than the clusters formed using DBSCAN.

The South Dakota dataset (Figure 10) consists of 236 contiguous polygons. Both the algorithms DBSCAN and P-DBSCAN were applied to this dataset using different values of ϵ , $MinPts$, and $MinPolys$.

The results of clustering using DBSCAN with different values of ϵ and $MinPts$ are shown in Figure 11. As before, we start with $\epsilon = 0.65$ which is the average distance between the polygons in the dataset. As $MinPts$ increases, there are polygons which are left unclustered (Figure 11(b) & 11(c)). As ϵ is increased to 0.75, all the polygons are clustered together in one

polygon. At this point, the value of $MinPts$ has no effect on the clustering process.

Fig. 10. Census Tract Polygons in South Dakota dataset

Fig. 11. Result of clustering using DBSCAN (a) $\epsilon = 0.65, MinPts = 1$ (b) $\epsilon = 0.65, MinPts = 2$ (c) $\epsilon = 0.65, MinPts = 5$ (d) $\epsilon = 0.75, MinPts = 5$

The results of clustering using P-DBSCAN with different values of ϵ and $MinPolys$, and $MinS = 4$ can be seen in Figure 12. As for DBSCAN, we start the results with $\epsilon = 0.65$ and $MinPolys = 1$ (Figure 12(a)) we see that all polygons are clustered with none of the polygons left unclustered. The number of clusters is more than DBSCAN, and none of the clusters contains only one polygon. When the value of $MinPolys$ is increased some of the polygons remain unclustered (Figure 12(b) & 12(c)). Finally, when $\epsilon = 1.0$ (Figure 12(d)), all the polygons are clustered together into one cluster. At this point, the value of $MinPolys$ has no effect on the clustering process.

Fig. 12. Results of clustering using P-DBSCAN (a) $\epsilon = 0.65, MinPolys = 1$ (b) $\epsilon = 0.65, MinPolys = 2$ (c) $\epsilon = 0.65, MinPolys = 5$ (d) $\epsilon = 1.0, MinPolys = 5$

The results obtained for DBSCAN and P-DBSCAN as shown above were compared using the compactness index. Once again we compare the results by computing

the average compactness index of all the clusters formed at a given ϵ and $MinPolys = 5$. Figure 13 shows the results. The number above each bar represents number of clusters. P-DBSCAN produces clusters with a lower compactness ratio, except for in one case where DBSCAN produces greater number of small clusters. This implies that the clusters formed using P-DBSCAN are more compact than the clusters formed using DBSCAN.

Fig. 13. Compactness Ratio for clusters formed using DBSCAN and P-DBSCAN.

C. Summary of Experiments

Summarizing the results of our experiments, we make the following conclusions:

- 1) ϵ plays a major role in deciding the formation of the clusters. The smaller the ϵ the smaller will be the clusters. As we increase ϵ , there will always be a value at which all the polygons will be grouped into one cluster. Further, depending on the average size of the polygons and thus the average distance between polygon centroids, the value of ϵ should be adjusted accordingly. That is, if the polygons are large, then ϵ should be increased, and vice versa.
- 2) $MinPolys$ parameter plays an important role in deciding if a polygon is a core polygon or not. Compared to $MinPts$ in DBSCAN, additional information could be derived from the average neighborhood of a cluster to better select a value for $MinPolys$. For example, if the polygons are mostly rectangular such that each polygon is likely to have 3 or 4 neighbors, then setting $MinPolys = 5$ might be too conservative, leading to many, small clusters. Further, by if the number of sectors of a polygon's neighborhood occupied by another polygon is generally large yet the number of neighboring polygons is low, then that indicates that polygons are surrounded by larger polygons. In that case, it might be more appropriate to set $MinPolys$ low.

V. CONCLUSION AND FUTURE WORK

We have proposed a new clustering algorithm for clustering polygons. Our algorithm is based on the density-based clustering algorithm DBSCAN. While

some concepts of DBSCAN are directly applicable for clustering polygons, concepts of core and border points as used in DBSCAN cannot be directly applied to define core and border polygons. Therefore, we re-define the concepts of core and border polygons. We introduce the concept of an outlier polygon, and a radial partition-based spatial neighborhood of a polygon which takes into account the topological properties of the polygons in addition to the density of the polygons in the dataset.

We also proposed using our modified Hausdorff distance function to compute the distance between the polygons while clustering them. Our distance function implicitly defines two polygons sharing a large extent of their boundaries to be close to each other. This is based on the intuitive concept of greater the sharing, more the similarity. However, we do not take into account that if the boundary is a country border, or a mountain range – a feature which may prohibit the clustering of the two polygons on either side together, then the distance should not be minimized. In our future research we will modify our distance function to take into account the type of the boundary between the two polygons.

Our comparison of the clustering results of DBSCAN and P-DBSCAN showed that more compact clusters are formed using P-DBSCAN. Thus our objective of producing compact clusters is satisfied by our proposed novel algorithm

Currently, the clustering is done only on the basis of distance between the two polygons. In our future experiments, we plan to introduce the concept of spatial autocorrelation in the process of clustering to enhance the compactness of the clusters further. We will be performing multi-dimensional clustering, where more attributes of the polygons will be taken into account while clustering the polygons.

ACKNOWLEDGMENT

We would like to thank Tao Hong for his help in writing the programs to pre-process the datasets. We would also like to extend our thanks to Dr. David Marx for his guidance.

REFERENCES

- [1] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic subspace clustering of high dimensional data for data mining applications" in *Proceedings of the ACM SIGMOD Conference*, Seattle, 1998, pp 94-105.
- [2] D. Fisher, "Knowledge acquisition via incremental conceptual clustering", *Machine Learning* 2(2), 1987, pp 139 – 172.
- [3] M. J. Egenhofer, and R. Franzosa, "On the equivalence of topological relations," *International Journal of Geographical Information Systems*, 9(2), 1994, pp 133-152.
- [4] M. J. Egenhofer, E. Clementini, and P. Di Felice, "Topological relations between regions with holes," *International Journal of Geographical Information Systems*, 8(2), 1994, pp 129-144.
- [5] M. J. Egenhofer, and D. M. Mark, "Modeling conceptual neighborhoods of topological line-region relations," *International Journal of Geographical Information Systems*, 9(5), 1995, pp 555-565.
- [6] M. Ester, A. Frommelt, H.-P. Kriegel, and J. Sander J., "Spatial data mining: database primitives, algorithms and efficient DBMS support," *Data Mining and Knowledge Discovery*, Kluwer Academic Publishers, 4, 2-3, 2000, pp 193-216.
- [7] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise", in *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD'96)*, Portland, OR, 1996, pp. 226-231.
- [8] J. Han, and Kamber, *Data Mining: Concepts and Techniques*, 2nd ed. Morgan Kaufmann Publishers, San Francisco, CA, 2006.
- [9] J. Han, M. Kamber, and A. Tung, "Spatial clustering methods in data mining: A Survey," in Miller, H., and Han, J., eds., *Geographic Data Mining and Knowledge Discovery*. Taylor and Francis. 2001, 21.
- [10] A. Hinneburg, and D.A. Keim, "An efficient approach to clustering in large multimedia databases with noise," in *Proceedings of 4th International Conference on Knowledge Discovery and Data Mining*, New York City, NY, 1998, 58 – 65.
- [11] D. Joshi, J. Zhang, A. Samal, and L.-K. Soh, "A distance function for polygon-based spatial cluster," *International Journal of Geographical Information System*, submitted for publication.
- [12] M. Ankerst, M.M. Breunig, H.-P. Kriegel, J. Sander, "OPTICS: Ordering points to identify the clustering structure," in *Proceedings of ACM SIGMOD International Conference on Management of Data*, Philadelphia, PA, 1999, 49-60.
- [13] G. Rote, "Computing the minimum Hausdorff distance between two point sets on a line under translation," in *Information Processing Letters*, 38, 1991, 123-127.
- [14] R.T. Ng, and J. Han, "Efficient and effective clustering methods for spatial data mining," in *Proceedings of 20th International Conference on Very Large Databases*, Santiago, Chile, 1994, 144 – 155.
- [15] S. Guha, R. Rastogi, and K. Shim, "CURE: an efficient clustering algorithm for large databases," in *Proceeding of ACM SIGMOD International Conference on Management of Data*, Seattle, WA, 1998, 73 – 84.
- [16] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, "Density-based clustering in spatial databases: the algorithm GDBSCAN and its applications," *Data Mining and Knowledge Discovery*, 2, 1998, pp. 169-194.
- [17] J. Schwartzberg, "Reapportionment, gerrymanders, and the notion of compactness." *Minnesota Law Review* 50 (1966) 443 – 452.
- [18] Sheikhholeslami, Gholamhosein, Surojit Chatterjee and Aidong Zhang, "WaveCluster: A Multi-Resolution Clustering Approach for Very Large Spatial Databases," in *Proceedings of the 24th Very Large Databases Conference (VLDB 98)*, New York, NY, 1998.
- [19] W. Wang, J. Yang, and R. Muntz, "STING: A Statistical Information Grid Approach to Spatial Data Mining," in *Proceedings of the 23rd Very Large Databases Conference (VLDB 1997)*, Athens, Greece. 1997
- [20] I.H. Whitten, and E. Frank, *Data Mining Practical Machine Learning Tools and Techniques (Second Edition)*, Morgan Kaufmann, San Francisco, 2005,
- [21] T. Zhang, Ramakrishnan, and M. Linvy, "BIRCH: an efficient data clustering method for very large databases," in *Proceeding ACM SIGMOD International Conference on Management of Data*, 1996, 103 – 114.