A Topological-Based Spatial Data Clustering

Hakam W. Alomari\textsuperscript{a}, Amer F. Al-Badarneh\textsuperscript{b}

\textsuperscript{a}Department of Computer Science and Software Engineering, Miami University, Oxford, OH, USA 45056; \textsuperscript{b}Computer Information Systems Department, Jordan University of Science and Technology, Irbid, Jordan 3030

ABSTRACT

An approach is presented that automatically discovers different cluster shapes that are hard to discover by traditional clustering methods (e.g., non-spherical shapes). This allows discover useful knowledge by dividing the datasets into sub clusters; in which each one have similar objects. The approach does not compute the distance between objects but instead the similarity information between objects is computed as needed while using the topological relations as a new similarity measure. An efficient tool was developed to support the approach and is applied to a multiple synthetic and real datasets. The results are evaluated and compared against different clustering methods using different comparison measures such as accuracy, number of parameters, and time complexity. The tool performs better than error-prone distance clustering methods in both the time complexity and the accuracy of the results.

Keywords: clustering, spatial data, knowledge discovery, data mining, topological relations

1. INTRODUCTION

Database systems have shown a great performance with stored data, but these systems might encounter some problems when dealing with space related and multidimensional data, like maps and satellite images [1]. In order to handle the spatial data, the database systems are augmented with spatial data types (e.g., points, lines, polygons, etc.) in its data model and query languages [2, 3, 4].

Spatial data may contain some useful (hidden) knowledge that tends to be large and complex in nature, thus the mining process must be automated in order to trace the progressively growth in the data. Clustering, as a mining task, divides the dataset into clusters (groups) in such a way, objects in one cluster similar to each other. The significant behind this process is to discover the natural distribution of the data, which leads to extract useful knowledge. The clustering problem has received a great attention within the past two decades; hundreds of research papers have been published presenting new clustering methods or improvement on existing methods with the aim of providing an optimal solution. Many applications took the benefit of clustering problems to improve the decision making process, such as information retrieval, text mining, spatial database applications, web analysis, medical diagnostics, pattern recognition, computational biology, and market research [5, 6, 7, 8]. Spatial database is a typical example of using clustering methods taking into consideration its huge quantity and complexity.

Amongst the different clustering techniques known in literature, hierarchical clustering models propose better versatility clustering spatial data, as they do not require an advance definition of the number of clusters to find. Based on the process of building clusters, hierarchical clustering methods are classified into two main approaches, agglomerative and divisive [5]. Divisive approach initially considers all data points as one big cluster, and then iteratively divides this cluster based on the similarity measure used between data points. On the other hand, agglomerative approach starts with each point as an individual cluster, and then depending on the measure type used as a similarity determinative, these points are grouped iteratively to build the final clusters.

Despite the type of clustering method used, the calculation of spatial clusters is, with few exceptions, based on the notion of a distance as a similarity measure, e.g., Euclidean Distance [9]. Unfortunately, calculating the distance matrix is quite costly in terms of computational time and space. As such clustering approaches generally do not scale well and while there are some (costly) workarounds, generating clusters for a very large dataset can often take days of computing time. Additionally, many tools are strictly limited to an upper bound on the dimensions of the dataset they can cluster.

\*Alomarhw@miamiOH.edu; phone 513 529-0356; \url{http://www.users.miamioh.edu/alomarhw/}
An accurate, fast, and scalable clustering approach is extremely useful for a number of reasons. Developers will have a very low cost and practical means to discover hidden patterns within reasonable time frames. This is very important for clustering new spatial objects and understanding how this object is related to other parts of the pattern. It will also provide an inexpensive test to determine if a full deep analysis of the data is warranted. Lastly, we think an accurate clustering approach could open up new avenues of research in mining useful knowledge based on clustering. That is, clustering can now be conducted on very large dimensional datasets and on more complex raw data in very practical time frames. This opens the door to a number of experiments and empirical investigations previously too costly to undertake.

We proposed a new spatial clustering approach called Agglomerative Clustering Using Topological Relations Effectively (ACUTE) and conducted a comparison study with different clustering methods from literature. The results show that the clusters produced by our approach are very reasonable with respect to accuracy. To demonstrate the scalability, we applied the tool to several synthetic and real datasets and present the results.

The remainder of this paper is organized as follows. Section 2 gives an overview of topological operators used in the proposed approach and an overview of related clustering. We present our new clustering method in Section 3. Section 4 gives the performance analysis and presents comparison with clustering algorithms. In Section 5, we summarize our paper and discuss possible future research directions.

2. TOPOLOGICAL OPERATORS AND CLUSTERING METHODS

In this section, we present an overview of the topological relations used in our approach and its modifications in order to identify the closeness between spatial points. Additionally, an overview of the related work of the basic types of clustering methods is also presented.

2.1 Topological operators

Topological relations [3, 18] between two spatial objects \( A \) and \( B \) can be specified using the well-known 9-intersection model \((3 \times 3 \text{ binary matrices})\). The regions that intersect to determine these nine relations are the interior, exterior, and boundary of objects \( A \) and \( B \). Table 1 shows the standard intersection matrix for objects \( A \) and \( B \) in the two-dimensional space. Where, °, −, and \( \partial \) symbols denote interior, exterior, and boundary relations, respectively. By allowing each matrix cell to be nonempty (equals to one) or empty (equals to zero) then we can consider \( 2^9 = 512 \) different topological relationships, in which only: disjoint, overlap, meet, contain, inside, equal, cover, and covered by relations can be established and represented in the two-dimensional space.

<table>
<thead>
<tr>
<th>( R(A, B) )</th>
<th>( A° \cap B° )</th>
<th>( A° \cap \partial B )</th>
<th>( A° \cap B° )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \partial A \cap B° )</td>
<td>( \partial A \cap \partial B )</td>
<td>( \partial A \cap B° )</td>
<td></td>
</tr>
<tr>
<td>( A− \cap B° )</td>
<td>( A− \cap \partial B )</td>
<td>( A− \cap B° )</td>
<td></td>
</tr>
</tbody>
</table>

Here we extend the original definitions of the topological relations in two ways. The first is adapting it to the spatial data points, and second is using three relations (i.e., disjoint, overlap, and meet) to build the final clusters. The assumption here that all spatial objects we are dealing with are singleton points has the same size in the Euclidean plan \( \mathbb{R}^2 \). Therefore, the relations: contains, inside, covers, and coveredBy are out of our interest. Moreover, we will not use the equal relation since we assume that every point has a unique position in the space.

We will now give a specification of the constraint rules (denoted Clustering Rules) that are generic for all type combinations considered. The clustering rules we used to cluster the spatial points do not take into account all the nine intersections of the standard intersection matrix. In total, an eight clustering rules are defined and proved in [19]. The three rules we are interested in are as follows:

**Clustering Rule 1 (Disjoint)** Two spatial points have a disjoint relationship, if the parts of one point intersect at most with the exterior of the other point, i.e., disjoint \((A, B) = A° \cap B° = \emptyset \land A° \cap \partial B = \emptyset \land B° \cap \partial A = \emptyset \land \partial A \cap \partial B = \emptyset \).

**Clustering Rule 2 (Meet)** Two spatial points have a meet relationship, if both interiors do not intersect, but the interior or the boundary of one point intersects the boundary of the other point, i.e., meet \((A, B) = A° \cap B° = \emptyset \lor (A° \cap \partial B \neq \emptyset \lor \partial A \cap \partial B \neq \emptyset)\).
our algorithm by computing the value of radii of the point agglomerate all points that overlapped and met with each other after increasing the radii of the points. Modifying the value analytically. Specifically, the proposed algorithm tries to determine a universal value of this value based on the dataset that is in clustering progress.

Here we will discuss how the proposed approach calculates the best value of the radius $r$, then how to approximate this value analytically. Specifically, the proposed algorithm tries to determine a universal value of $r$ that could be applied to any dataset (regardless the data distribution in the spatial space), or in the worst-case find a way to dynamically calculate this value based on the dataset that is in clustering progress.

All existent clustering algorithms specifying the closeness between spatial objects using a distance similarity measure, so they assume that similar objects closest to each other forming a cluster. Our proposed algorithm utilizes this notation by agglomerate all points that overlapped and met with each other after increasing the radii of the points. Modifying the radii of the points with incorrect value of $r$ can lead to extract inaccurate number and odd shapes of clusters. We start our algorithm by computing the value of $r$ experimentally using a dataset with previously well-known true clusters. The

**Clustering Rule 3 (Overlap)** Two spatial points have an overlap relationship, if the interior of each point intersects both the interior and the exterior of the other points, i.e., $overlap\ (A, B) = A^\circ \cap B^\circ \neq \emptyset \land A^\circ \cap B^\circ \neq \emptyset \land B^\circ \cap \neg A \neq \emptyset$.

### 2.2 Spatial clustering methods

Clustering analysis is a commonly used approach for understanding and detecting hidden patterns in the raw data. The idea is quite simple, given a dataset’s object and the location of this object in the space, tell me what other objects of the dataset are affected (similar) by this object. The approach has been used successfully for many years for various real world applications [8, 20, 21, 22]. For example, clustering was used to help address image coding clustering, document clustering, and spatial clustering in geo-informatics. These various applications of clustering require different properties; thus, a number of different definitions have been proposed [14, 15, 17].

The two main abstract definitions are the spatial and non-spatial clustering approaches. These definitions are covered in detail in various surveys of the clustering literature [5, 12, 23, 24, 26, 27]. Interestingly, each survey presents the definitions from a slightly different perspective. For example, some references focus mainly on the applications of clustering techniques. Other references compare different implementations and classify the clustering techniques according to their empirical results. Finally, some authors compare and classify clustering techniques in order to identify relations between them.

Although there are a number of similarities between spatial and non-spatial clustering algorithms, spatial datasets/databases in particular need specific requirements (e.g., identify irregular shapes, sensitivity to noise, higher dimensionality, etc.) that make special needs for clustering algorithms. There are a number of different methods proposed for performing spatial clustering but the four main divisions are hierarchical, partitional, grid-based, and density-based clustering. $k$-means [10, 11] is one of the most famous and simplest partitional clustering algorithms. It defines the cluster as similar objects to the centroid of the cluster, which is usually the mean of a group of objects. $k$-means starts by choosing $k$ initial centroids, where $k$ is user-predefined number of clusters, and then iteratively updates the centroids until there is no further change with these centroids. After that a number of different clustering techniques and tools have been proposed and implemented. These techniques are broadly distinguished according to the similarity measure used and the input requirements of the clustering method.

### 3. TOPOLOGICAL CLUSTERING

This section aims at proposing a new clustering algorithm called Agglomerative Clustering Using Topological Relations Effectively (ACUTE). ACUTE algorithm supersedes existing clustering algorithms in two main aspects: one its validity for all cluster shapes, given that it agglomerates points in all directions. Second, it reduces both the number of parameters and comparisons that is needed by most clustering algorithms so as to perform the clustering process.

#### 3.1 Algorithm overview

The algorithm initially computes the new appropriate value of radius $r$ experimentally, then iteratively selects a point with flag field equal to zero from the point’s table and scans the database in order to compare the current point with other points. The points that related with overlap and meet relationships are belong to the same cluster and the algorithm sets its flag fields to one, so no more comparison with it. The topological relations are applied by comparing the corresponding dimensions of points $A$ and $B$, so if the relation between the two points is overlap or meet then these points belongs to the same cluster. Finally, the final clusters are identified while scanning (sequentially) the cluster table, then if the final cluster just has a single point, or a user pre-defined threshold, then this cluster is marked as an outlier.

#### 3.2 Radius approximation

Here we will discuss how the proposed approach calculate the best value of the radius $r$, then how to approximate this value analytically. Specifically, the proposed algorithm tries to determine a universal value of $r$ that could be applied to any dataset (regardless the data distribution in the spatial space), or in the worst-case find a way to dynamically calculate this value based on the dataset that is in clustering progress.
The algorithm starts by clustering the data points initially using a small value of $r$ (e.g., 0.1). After that the algorithm iteratively increases the value of $r$ and cluster the objects. This process continues until the accuracy value (true ratio) starts to degrade or reaches 100% (which occurs first).

Essentially, the process of increasing the value of $r$ in its core is considered a distance calculation process. Hence, for a faster computation of radius, we can use one of the well-known approximation methods such as $(1+\varepsilon)$-approximate nearest neighbor [28] to reduce the computational cost to only $O(f \log n)$, where $f$ is a factor depending only on dimension. The proposed algorithm here divides the clustering process into two stages. In the first stage, the algorithm use a cheap distance measure to create overlapping circles (or hypersphere for dimensions greater than two). The radius $r$ of any circle is equal to the half of maximum distance of minimum pair wise distances. Significantly, a data point may appear under more than one circle, and every data point must appear in at least on circle. Note that every circle covers a data point in its center with a radius equal to $r$, the value of $r$ is obtained from the distance matrix, and thus the proposed algorithm does not require any input parameter. In the second stage, the proposed algorithm applies the topological relations. The values of overlap, meet, and disjoint relations are obtained from the overlapped circular regions using the equation $2r - \text{dist}(p, q)$ for a given pair of point's $p$ and $q$. The proposed algorithm partitions data space into overlapped circles. The radius $r$ of these circles should be depends on the far nearest neighbor distance of data space, since we want to make sure that the circle (of radius $r$) around the farthest point will meet with the circle around the nearest neighbor of this point.

### 3.3 Accuracy Calculation

Accuracy value measures the portion of correct cases to all cases. In order to calculate the accuracy, the true clusters of points should be known in advance. Then the correct number of cases can be calculated by comparing the resulted clusters with existing true ones. So the accuracy will be: $\text{Accuracy} = \left( \frac{\text{number of correct cases}}{\text{total number of cases}} \right) \times 100$.

![Figure 1](http://proceedings.spiedigitallibrary.org/)

Figure 1. The clusters representations after each change in the $r$-value. The true clusters for this particular example are seen in (d) in (d) when the $r$-value is equal to 0.5.

Figure 1 shows how the sizes of the points are influenced with the increases in the $r$-value. The significant behind this step is to observe the differences in cluster shapes while the radius is changed. As shown in Figure 1(a), all points initially represented with $r$-value by default equals to $\sim 0$. Then by increasing the value of $r$ gradually, the natural clusters of the dataset start to appear. The process of increasing the value of $r$ continues until either the resulted clusters (after each increment) equal to the one in the true cluster table or the cluster’s accuracy value starts to degrade.

Figure 1(d) represents the final clusters when the $r$-value equals to 0.5, in this case we have two clusters and one outlier. Assume we continue increasing the value of $r$, and then we have cases as those shown in in Figure 1(e) and Figure 1(f). The points start to merge between each other, so the accuracy value starts to decrease. One step previous of the decreasing step is comprises the best $r$-value that we should consider to cluster the dataset.
4. EXPERIMENTS AND PERFORMANCE ANALYSIS

Different real and synthetic datasets with different sizes, densities, and shapes were used to evaluate the ACUTE performance and accuracy. We used four well-known real datasets, which are: Iris, Wine, Congressional Votes, and Mushroom, in addition to six synthetic datasets which are: Hepta, Density, ACUTE, Forty, Twenty, and Spiral. The characteristics of the datasets are summarized in Table 2.

Table 2. Four real and six synthetic datasets properties. The size of the clusters for the first two datasets (Iris and Wine) are already known.

<table>
<thead>
<tr>
<th>Data</th>
<th>Number of Dimensions</th>
<th>Number of Clusters</th>
<th>Outliers</th>
<th>Number of Points</th>
<th>Cluster Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>4</td>
<td>3</td>
<td>No</td>
<td>150</td>
<td>50, 50, 50</td>
</tr>
<tr>
<td>Wine</td>
<td>13</td>
<td>3</td>
<td>No</td>
<td>178</td>
<td>71, 59, 48</td>
</tr>
<tr>
<td>Cong.Votes</td>
<td>16</td>
<td>6</td>
<td>Yes</td>
<td>712</td>
<td>1000</td>
</tr>
<tr>
<td>Mushroom</td>
<td>22</td>
<td>24</td>
<td>Yes</td>
<td>8124</td>
<td>1000</td>
</tr>
<tr>
<td>Hepta</td>
<td>3</td>
<td>7</td>
<td>No</td>
<td>212</td>
<td>1000</td>
</tr>
<tr>
<td>Density</td>
<td>2</td>
<td>3</td>
<td>No</td>
<td>65</td>
<td>1000</td>
</tr>
<tr>
<td>ACUTE</td>
<td>2</td>
<td>5</td>
<td>No</td>
<td>311</td>
<td>1000</td>
</tr>
<tr>
<td>Forty</td>
<td>2</td>
<td>40</td>
<td>No</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>Twenty</td>
<td>2</td>
<td>20</td>
<td>No</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>Spiral</td>
<td>2</td>
<td>2</td>
<td>No</td>
<td>1000</td>
<td>1000</td>
</tr>
</tbody>
</table>

The Iris dataset is considered as one of the most popular datasets in clustering literature. The Congressional Votes dataset it is the United States Congressional Voting Records in 1984. Each record represents one congressman’s votes on 16 issues (attributes). A classification label of republican or democrat is provided with each record. This dataset contains 435 records with 168 republicans and 267 democrats. Mushroom dataset has 22 attributes with 8124 records. Each record has the physical characteristics of a single mushroom. The Wine dataset are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. This dataset had around 13 variables, and in the clustering context, this is a well-posed problem with well-behaved class structures. It is considered as a good dataset for testing a new clustering algorithm. For a complete description, all the datasets can be downloads from Machine Learning Repository at University of California, Irvine [25].

4.1 Evaluation criteria

Here we evaluate the clustering results of our tool to determine if correct clusters are produced, and are produced efficiently. The time and cost it takes to generate the cluster, including execution time and number of parameters, is of particular interest with respect to usability of the method. In addition, we want to determine if the results obtained by ACUTE are comparable to others in terms of accuracy. However, since the implementations of these tools have so few aspects in common, it is not meaningful to compare all of the relevant aspects of the different implementations. Therefore, we focus our attention on evaluating clusters of both tools, by taking into consideration the correctness, size of the results, time, and the limitations of both tools.

4.2 Performance analysis

The overall computational complexity of ACUTE depends mainly on the number of comparisons between data points. Increasing the r-value gradually degrades the number of comparisons between data points, from the time when the number of points that overlap or meet is increased. In other words, there is no need to compare the points that already belongs to another clusters. The amount of time required by the comparison process depends mainly on (1) the number of data points in the dataset, and (2) the value of radius r.

The worst-case complexity is obtained when each data point considered as a single cluster. That is, when the r-value is initially set to very small value (~ zero). This corresponds to the worst case because the initial chosen data point is compared with number of points equal to n – 1, the second chosen point is compared with n – 2 points, these comparisons is continue to decrease by one until the final chosen point is compared only with one data point. The amount of time needed to compare n data points is (n² – n)/2 which is O(n²). On the other hand, when the r-value is increased gradually, the number of comparisons is decreased gradually until it is equal to n – 1, in this case all the points are reached from the initial data point.
As an example, Figure 2 shows the execution time of ACUTE using Hepta dataset, with different r-values. As seen, in general as the value of r increases, the execution time of ACUTE decreases due to the decreases in the total number of comparisons between data points. However, the execution time and the accuracy values (ACC) are sometimes stable on different values of r-value while the r-value is still increasing. This is because the number of comparisons is stable too.

Table 3. The accuracy values (ACC) versus the radii values (r-value) on the four real datasets (Wine, Density, Cong. Votes, and Mushroom).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>ACC</th>
<th>r-value</th>
<th>Number of iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wine</td>
<td>79.8</td>
<td>47.5</td>
<td>20</td>
</tr>
<tr>
<td>Density</td>
<td>100</td>
<td>14</td>
<td>8</td>
</tr>
<tr>
<td>Cong. Votes</td>
<td>92.1</td>
<td>32</td>
<td>15</td>
</tr>
<tr>
<td>Mushroom</td>
<td>100</td>
<td>11.5</td>
<td>53</td>
</tr>
</tbody>
</table>

4.3 Experimental results

In order to facilitate choosing the appropriate value of radius r, and to indicate the impact on the clustering results with different values of r, Figure 3 illustrates the execution of the ACUTE algorithm on Hepta, Iris, and Spiral datasets. This figure shows the exchange in the accuracy values (ACC) when the radius value is increased. The Hepta and the Spiral datasets hit an accuracy value equal to 100% (first hit) when the r-value is equal to 0.5 and 0.3, respectively. In another hand, the ACC value on the Iris dataset at best case is equal to 86.7% at r-value equal to 0.4. Those three datasets are plotted together in the same figure, as the scale values on the r-value are equals.

For the sake of space and simplicity the results of the Wine, Cong. Votes, Mushroom, and Density datasets are shown in Table 3. The ACUTE algorithm hit a 100% accuracy value on both Density and Mushroom datasets with r-values equal to 14 and 11.5, respectively. However, using the Wine and Cong. Votes datasets, the accuracy values (in the best case) are equal to 79.8 and 92.1, respectively. The number of iterations column represent the number of increases of the r-value till reach the highest ACC value. This number is changed dynamically if the amount of increase in the r-value is larger. For example, in the Wine dataset the amount of increase in the r-value is equal to 2.5 units starting from zero, therefore the number of iteration is equal to 20 to reach r-value equal to 47.5.

ACUTE iteratively clustering the data after increasing the value of r, after that the ACC value is calculated. This process is continuing until no more enhancements can be done on the accuracy value. For example, as shown in Figure 3 as the r-value increases the accuracy value increases too, since the number of points that overlapped or meets is increased. This enhancement on the accuracy value start to degrades then become more stable when the r-value increased to be very large, as a result all clusters are merged together. In other words, the r-value considered being large when all the points in the space are merged in one cluster (i.e., first cluster). In such a case, the accuracy value is stable and equal to (the number of points in the first cluster / total number of points in the dataset). As an example for this case, the Iris dataset...
has three clusters each one has 50 points, so when we enlarge the r-value too much then the accuracy value unchanged and it is equal to (50 points in the first cluster / 150 total dataset points = 0.33). As we can observe from the figure of the accuracy values above, when the r-value increased the accuracy value increased, then the accuracy value either stable in different iterative and (or) related inversely with the r-value then it is stable.

Figure 3. ACUTE’s Accuracy values (y-axis) on Hepta, Iris, and Spiral datasets with different number of radii (x-axis).

Figure 4. Percentage of clustering algorithms using ACUTE dataset.

Figure 4 illustrates the true point’s accuracy values of the ACUTE, k-means, k-medoids [13], DBSCAN [16], and CLIQUE [26] clustering algorithms based on the ACUTE dataset. As seen, ACUTE accuracy value on ACUTE dataset is 100% with r-value equal 13, and k-means 66.23% with k equal 5. In addition, wrong choice of parameters on the CLIQUE degrades its accuracy value to 32.15%. In addition, we illustrate a comparison between the ACUTE clustering algorithm and other clustering algorithms from the number of parameters needed to accomplish the clustering process. ACUTE like most of other tools, needs one parameter to accomplish the clustering process, this is the r-value.

5. CONCLUSIONS AND FUTURE WORK

The main contribution of this paper was a new spatial clustering algorithm. The proposed algorithm derives benefit from the topological relations as a spatial object’s similarity measure in order to overcome the extra overhead calculations generated when using the distance functions. Moreover, the algorithm builds clusters without known information of any calculated values in the dataset such as mean, median, and standard deviation. Proposed algorithm detects the natural clusters in the dataset with different cluster shapes. It can deal with points as a representative of all spatial data types, and can merge the points in all directions.

We have compared the ACUTE algorithm with other clustering algorithms using different datasets with different sizes, shapes, densities, and number of dimensions. Experimental results have shown that our algorithm reduces the clustering processing time, in the best case, to O(n – 1) by reducing the number of comparisons in the building clusters. Additionally, ACUTE algorithm clusters most of the datasets with high 100% of accuracy value.
The ACUTE algorithm specifies $r$ by iteratively increase its value until the accuracy value starts to degrade. As a future work we suggest that the $r$-value should be calculated analytically, we suggest benefiting from the distribution of the dataset in the space as a good indicator of the $r$-value. However, every dataset’s space can take different value of $r$.

REFERENCES


