CTVN: Clustering Technique Using Voronoi Diagram

P S Bishnu[1] and V Bhattacherjee[2]
[1,2]Department of Computer Science and Engineering
Birla Institute of Technology, Ranchi, India
Email: [1]psbishnu@gmail.com, [2]vbhattacharya@bitmesra.ac.in

***Abstract*-Clustering technique is one of the most important and basic tool for data mining. We propose a new clustering technique using *K*-Means algorithm and Voronoi diagram to unfold the hidden pattern in a given dataset. In the first phase we use *K*-Means algorithm to create set of small clusters and in the next phase using Voronoi diagram we create actual clusters. *K*-Means and Voronoi diagram based clustering results are of high quality and robustness and are able to identify noise. Our proposed algorithm is tested on synthetic dataset and results are presented.**

Index Terms—Data Mining, *K*-Means algorithm, Voronoi diagram

I. Introduction

Clustering is the method of grouping the data into sets so that data points within the set should have high similarity while those within different sets are dissimilar [1] [2]. *K*-Means is the popular partitioning technique, which attempt to decompose the data set of disjoint clusters. This is simple, robust and time efficient techniques. The *K*-Means algorithm has some inherent drawbacks, so various improvement techniques are developed [10]. This paper presents a new clustering technique using *K*-Means and Voronoi diagram to find the actual clusters. We call our new algorithm CTVN (which is anagram of the bold letters in **C**lustering **T**echnique Using **V**oro**N**oi Diagram). The orientation of the paper is as follows: Section 2 presents survey of the related work. Section 3 gives an overview of the *K*-Means and Voronoi diagram. Section 4 presents our proposed CTVN algorithm. Next we discuss results in Section 5 and conclusions in Section 6.

II. Related Work

Various techniques have proposed for clustering using Voronoi Diagram [3] [4] [5]. Koivistoinen et al. suggested an agglomerative clustering algorithm which accesses density information by constructing a Voronoi diagram for the input dataset. They use Voronoi diagram for singleton cluster selection. Without measuring any distance, the initial clusters are merged together with neighboring cells as long as the cell volumes are below a user defined threshold value, since only neighboring cells are candidates for merging and the Voronoi diagram readily provides us with the neighborhood information. On the other hand, this technique needs a threshold volume to limit merging of cells [6]. Yan et al. proposed an algorithm for point cluster generalization, considering the transmission of statistical, thematic, topological, and metric information, based on measures for quantifying these types of information and on strategies that integrate the measures into the algorithm. Owing to the use of the Voronoi diagram the algorithm is parameter free and fully automatic [7].

III. Overview of *K*-Means and Voronoi diagram

A K-Means clustering algorithm

The *K*-Means algorithm is based on squared error minimization method. We discuss the *K*-Means algorithm as follows:

Table 1: *K*-Means algorithm

```
Step 1: Randomly choose $k$ data points from the whole data set as
        initial cluster centers.
Step 2: Calculate Euclidean distance of each data point from each
        cluster center and assign the data points to its nearest
        cluster center.
Step 3: Calculate new cluster center so that the squared error
        distance of each cluster should be minimum.
Step 4: Repeat step 2 and 3 until the cluster centers do not
        change.
Step 5: Stop the process.
```

The Time complexity of *K*-Means algorithm is $\mathrm{O}(nkt)$ where n is number of data objects, k is number of clusters and t is number of iterations.

B Voronoi Diagram

In the field of computational geometry Voronoi diagram is the one of the most important and useful technique. Given a D set of n data points o_1, o_2, o_3, …,o_n in a plan, the Voronoi diagram (Vor(D)) is a subdivision of the plane into Voronoi cells (V(o_i) for o_i) (Figure 1). The Voronoi cells are the set of points u that are closer or as close to o_i than to any other point in D. i.e. $V(o_i) = \{r | d(o_i,u) \le d(o_j,u) \forall j \ne i\}$, where d is the Euclidian distance. The Voronoi diagram divides the plane into n convex polygon regions (for each o_i), the vertices (P

of Figure 1) of the diagram are the Voronoi vertices, and the borders between two adjacent Voronoi cells are called as Voronoi edges (*E* of Figure 1) [8] [9]. Note that each Voronoi vertex is the center of a circle touching three or more data points lying in its adjacent Voronoi cells [8] (Figure 1)

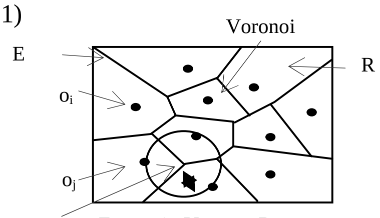


Figure 1 Voronoi Diagram

IV. The proposed CTVN algorithm

The CTVN algorithm begins by applying the *K*- Means algorithm upon the dataset to return set of centroids (Q_m) , where $C<m<<n$, *C* is actual numbr of clusters, *n* is number of data points and *m* is the user defined number. (Step 1 of Table 2). Voronoi diagram is created on the set of centroids to get the set of vertices (*P*) which are labeled as unmarked (Step 2 and 3 of Table 2). These vertices are sorted by their x-axis co-ordinates (Step 4 of Table 2) and *subcluster_id* variable *c* is set to one (Step 5 of Table 2). Steps 6 and 8, vertices are chosen one at a time. Note that each vertex is the centre of the circle touching three or more centroids lying in adjacent Voronoi cells. For each vertex, this circle is created and if its radius is less than a threshold γ then choose any marked centroids with smallest *subcluster_id* in the circle and mark other centroids (marked or unmarked) to this *subcluster_id*. Update the subcluster array to reflect these changes. If no marked centroid is found, make the unmarked centroids with the *subcluster_id* *c*. The steps 6 to 8 are repeated for all vertices. This creates the set of subclusters of centroids. All unmarked are noise centroids. Data points are assigned to the nearest centroids and this creates the actual clusters or noise. The steps of our proposed CTVN algorithm are given below:

Table 2: CTVN Algorithm

```
1. apply $K$-Means algorithm on given dataset ($D$),return set of
   centroids ($Q$);
2. create Voronoi diagram on set of centroids ($Q$), return set of
   vertices ($P$);
3. set all the centroids ($Q$) as unmarked;
4. sort the vertices by x-axis;
5. set $c$=1;            // subcluster_id
6. choose vertices ($P_i$) one by one in increasing order of x axis;
7. create circle such that the circumference of the circle touches
   centroids ($Q_k$, $k$>=3), where $P_i$ is center of that circle;
8. if the radius ($R_i$) is less than $\gamma$ then
       if any centroid ($Q_k$) is already assigned a subcluster_id ($c$)
            then assign the unmarked (or readjust the mark)
            centroids to the smallest subcluster_id ($c$);
            update subcluster array $s$ to readjust the subcluster_id
            ($c$);
       else
                 assign subcluster_id = $c$ on unmarked centroids;
                 increment $c$;
           set centroids as marked;
9.  continue steps 6 to 8 till all the vertices ($P$) processed;
10. return set of subcluster_id ;
11. all unmarked centroids are noise centroids;
12. assign data points ($D$) on nearest centroids;
13. return clusters ($C$) and noise ($N$);
```

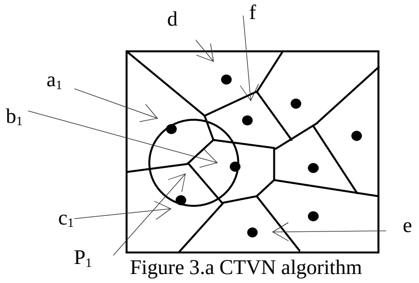


Figure 3.a CTVN algorithm

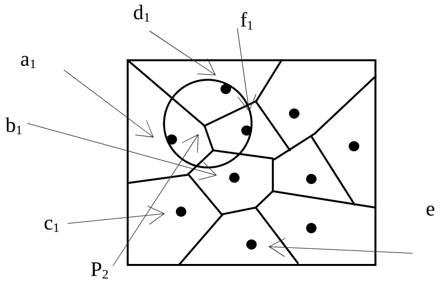


Figure 3.b CTVN algorithm

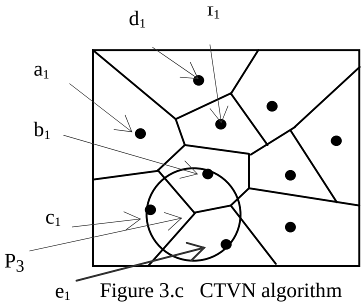


Figure 3.c CTVN algorithm

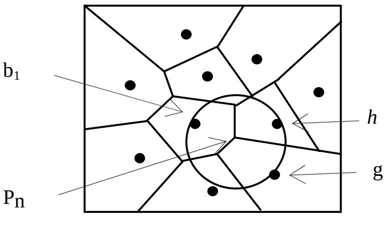


Figure 3.d CTVN algorithm

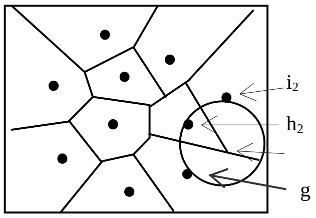
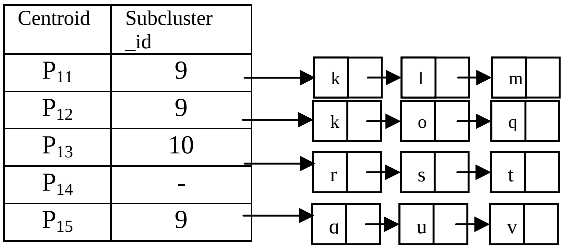
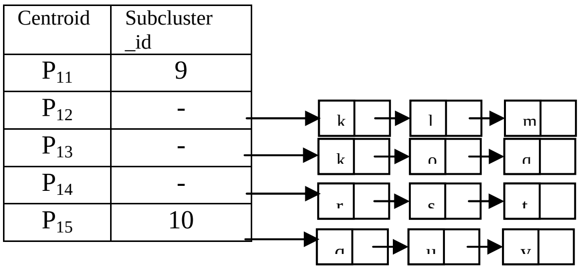


Figure 3.e CTVN algorithm



Centroid	Subcluster _id
P_{11}	9
P_{12}	9
P_{13}	10
P_{14}	-
P_{15}	9

Figure 4.a Adjustment of *subcluster_id*



Centroid	Subcluster _id
P_{11}	9
P_{12}	-
P_{13}	-
P_{14}	-
P_{15}	10

Figure 4.b Adjustment of *subcluster_id*

Illustration

Figure (3.a) to (3.e) illustrates the working of the CTVN algorithm. In Figure (3.a) the vertex P_1 is chosen and its circle (less than γ in radius) touches centroids *a*, *b*, *c*. These are marked as *subcluster_id* one. Next P_2 is chosen in Figure (3.b) and centroids *a*, *d*, *f* which also get marked with *subcluster_id* one, because of *a*. Similarly in Figure (3.c) *b*, *c*, *e* get marked with *subcluster_id* one (because of *b* which is already marked as one). In Figure (3.d), P_n is chosen and this circle touches centroids *b*, *h*, *g* out of which *b* is marked by *subcluster_id* one. However *h*, *g* remain unmarked because its circle radius greater than γ. Later *h*, *g* get marked with *subcluster_id* two when vertex P_f is chosen. Figure (4.a) and (4.b) illustrates the adjustment of *subcluster_id* (step 8 of Table 2, CTVN algorithm) say, vertex $P_{11,}$ P15, P_{12} are chosen one after another. When P_{11} is chosen, *k*, *l*, *m* get marked with *subcluster_id* nine and when P_{15} is chosen *q*, *u*, *v* get marked with *subcluster_id* ten (Figure (4.a)). Now when P_{12} is chosen its circle touches *k*, *o*, *q* out of which *k* and *q* are already marked and *k* has the smallest *subcluster_id* (nine). Hence P_{12} get marked by *subcluster_id* nine as shown in Figure (4.b). Further, vertex P_{15} also gets adjusted to *subcluster_id* nine. Next when, say, vertex P_{13} is chosen, it gets marked with *subcluster_id* ten, and so on.

V. RESULTS

The CTVN algorithm was validated upon four synthetic datasets. Our results were compared with that of *K* Means algorithm (best validity values for each dataset) and are presented in Table 3. The CTVN algorithm is able to identify actual number of clusters for all the datasets and validity values [11] obtained are minimum.

Table 3: Results

Dataset	Actual numbers of cluster	After using *K* Means Algorithm		After using proposed CTVN Algorithm	
		Validity	No. of clusters	Validity	No. of Clusters
Dataset1	3	1.88	3	0.98	3
Dataset2	2	4.78	3	3.77	2
Dataset3	3	5.23	3	5.00	3
Dataset4	4	1.89	3	1.01	4

VI. CONCLUSION

In this paper Voronoi diagram have been used in conjunction with *K* Means algorithm for identifying hidden patterns in a dataset and create actual clusters. Further, noise data points are also identified by our CTVN algorithm. Results of Table 3 show that our algorithm performs very well. Further, user defined parameters for our algorithm is only two; hence it is less user dependent. Our future work comprises of validation of the proposed algorithm on real dataset for scalability and robustness.

REFERENCE

[1] Jiawei Han and M. Kamber, *Data mining concepts and techniques*, Morgan Kaufmann Publishers, 2007.

[2] M H Dunhum, *Data Mining- Introductory and Advanced Topics*, Pearson Education, 2007.

[3] Ickjai Lee, "Voronoi-based Topological Hierarchical Clustering", *International Conference on Computational Intelligence for Modeling, Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, Vol 2, pp-484-489, IEEE, 2005.

[4] Ben Kao, Sau Dan Lee, David W. Cheung, Wai- Shing Ho, K. F. Chan, "Clustering Uncertain Data using Voronoi Diagrams", *Eighth IEEE International Conference on Data Mining*, pp.- 333-342, 2008.

[5] Thomas Schreiber, "A Voronoi Diagram Based Adaptive *K*-Means-Type Clustering Algorithm for Multidimensional Weighted Data", *Computational Geometry Methods, Algorithms, and applications*, Springer Link, pp.- 265-275, 2006.

[6] Heidi Koivistoinen, Minna Ruuska, and Tapio Elomaa, "A Voronoi Diagram Approach to Autonomous Clustering", *Springer-Verlag Berlin Heidelberg* , Vol.- 4265, pp- 149-160, 2006.

[7] Haowen Yan, Robert Weibel, "An algorithm for point cluster generalization based on the Voronoi diagram", *Computers & Geosciences* 34, pp. 939–954, 2008.

[8] Mark de Berg, Otfried Checong, Marc Van Kreveld, Mark Overmars, Computational Geometry Algorithms and Application 3rd Edition Springer publication.

[9] Jilin Qu, "Outlier detection using voronoi diagram", *International symposium on computational intelligence and design*, Vol 1, pp.- 495-498, IEEE, 2008.

[10] P.S.Bishnu and V Bhattacharjee, "A New Initialization Method for K-Means Algorithm using Quad Tree",*NCM2C*, 2008, JNU, New Delhi.

[11] Noha A. Yousri, Mohamed S. Kamel, Mohamed A. Ismail, " A novel validity measure for clusters of arbitrary shapes and density", *ICPR-08, 19th International conference on pattern recognition*, pp 1-4, IEEE, 2008.