# Efficient data clustering algorithm designed using a heuristic approach

Poonam Nandal, Deepa Bura* and
Meeta Singh

Department of Computer Science and Engineering,
Faculty of Engineering and Technology,
Manav Rachna International Institute of Research and Studies, India
Email: poonamnandal.fet@mriu.edu.in
Email: deepa.fet@mriu.edu.in
Email: meeta.sangwan@gmail.com
*Corresponding author

**Abstract:** Information retrieval from a large amount of information available in a database is a major issue these days. The relevant information extraction from the voluminous information available on the web is being done using various techniques like natural language processing, lexical analysis, clustering, categorisation, etc. In this paper, we have discussed the clustering methods used for clustering of large amount of data using different features to classify the data. In today's era, various problem solving techniques makes the use of a heuristic approach for designing and developing various efficient algorithms. In this paper, we have proposed a clustering technique using a heuristic function to select the centroid so that the clusters formed are as per the need of the user. The heuristic function designed in this paper is based on the conceptually similar data points so that they are grouped into accurate clusters. *k*-means clustering algorithm is majorly used to cluster the data which is also focussed in this paper. It has been empirically found that the clusters formed and the data points which belong to a cluster are close to human analysis as compared to existing clustering algorithms.

**Keywords:** clustering; natural language processing; *k*-means; concept; heuristic; Euclidean distance; 2D algorithm; information retrieval; Manhattan distance; density concept.

**Reference** to this paper should be made as follows: Nandal, P., Bura, D. and Singh, M. (2021) 'Efficient data clustering algorithm designed using a heuristic approach', *Int. J. Data Analysis Techniques and Strategies*, Vol. 13, Nos. 1/2, pp.3–14.

**Biographical notes:** Poonam Nandal received her Bachelor of Engineering in Information Technology in 2005 from the Institute of Technology and Management affiliated to Maharishi Dayanand University Rohtak, and Master of Technology in Computer Science and Engineering in 2009 from the Career Institute of Technology and Management affiliated to Maharishi Dayanand University, Rohtak. She completed her PhD in 2017 from the YMCA University of Science and Technology in the field of Natural Language Processing. She has 11 years of teaching experience. Presently, she is working as an Associate Professor in the Department of Computer Science and Engineering, Faculty of Engineering and Technology, Manav Rachna International Institute of Research and Studies, Faridabad. Her areas of interest include information retrieval, semantic web, artificial intelligence, natural language processing, soft computing, and compiler design.

Deepa Bura received her Bachelor of Engineering in 2002 from the Vaish College of Engineering affiliated to Maharishi Dayanand University Rohtak, and Master of Technology in Information Technology in 2009 from the University School of Information Technology affiliated to Guru Gobind Singh Indraprastha University, Delhi. She completed her PhD in 2018 from the Uttarakhand Technical University in the field of Software Engineering. She has 16 years of teaching experience. Presently, she is working as an Assistant Professor in the Department of Computer Science and Engineering, Faculty of Engineering and Technology, Manav Rachna International Institute of Research and Studies, Faridabad. Her areas of interest include data mining, software engineering, cloud computing and soft computing.

Meeta Singh received her Master of Technology in Information Technology from the Guru Gobind Singh Indraprastha University, New Delhi in 2007. She completed her PhD in Computer Science and Engineering from the Bhagwant University, Ajmer in 2015 in the field of mobile ad hoc networks. She has 18 years of teaching experience. Presently, she is working as an Associate Professor in the Department of Computer Science and Engineering, Faculty of Engineering and Technology, Manav Rachna International Institute of Research and Studies, Faridabad. Her areas of interest include cloud computing, wireless ad hoc network, computer architecture, microprocessor and data mining.

## 1 Introduction

Data mining is the process of analyse the relevant data from different extents. Data mining is basically the retrieval of relevant information from the collection of information in the form large databases. It also focuses the change of retrieved data into understandable and readable manner. It is a basic process of exploring the data from different areas to use it for various purposes like data managing in big areas.

It has been observed that data mining is the best way of managing the existing data by means of software and hardware platforms to escalate the worth of current data by getting new various types of associated information of the datasets. Basically, the techniques used for mining the data have extensive long processes of researches done over it. Data mining technology can also produce novel techniques further used in businesses with the help of various tools and techniques. This huge amount of data needs to be classified by various data classification techniques that can be supervised or unsupervised. One of the most widely used techniques is to classify the dataset into different clusters commonly referred to as clustering. Clustering is grouping of similar data or information. It is the most commonly and widely used domain for discovery of knowledge in today's world. It is the most fundamental technique of data mining.

Data mining is the innovation of exciting, unpredicted or valuable structures in huge datasets. Basically, it has two rather dissimilar aspects. The first one constitutes large-scale, 'global' structures. The major aim of global structures is related to modelling of shapes, or extraction of features of the shapes, having various distributions for normalisation. The second one constitutes the small-scale, 'local' structures. The main aim of local structures is basically to detect the anomalies present in the huge amount of data and further decides that the occurrence of the data is real or chance occurrences. In the framework of signal recognition available in the pharmaceutical sector, major interest lies in the second aspect of the above mentioned two aspects.

Clustering is a data classification technique which is based on unsupervised learning. It is basically grouping of similar data which is in addition used for exploring data. There are so many applications of clustering in numerous domains like data mining, biological sciences, pattern recognition document retrievals. It's most common application used by various MNCs, industries and companies to manage their data according to their requirements. This paper discusses the different comprehensive techniques of clustering which is one of the foremost tasks for analysis of data mining. It recognises related records groups that can be used as an original point for discovering additional multiple data about that. This technique helps us to solve large calculation problem to maintaining or managing data such as graphic-based. We use various additional analyses of different techniques of data mining that gives clusters of similar types of data. There are various methods of clustering and each method is chosen according to the desired output. Clustering can be categorised as follows: hierarchical method, spectral clustering, grid-based clustering, grid-based method, density-based, partitioning method (Maulik and Bandyopadhyay, 2002).

Hierarchical clustering algorithm data is classified in the form of a tree. It is further categorised in two types, i.e., agglomerative and divisive. The agglomerative hierarchical clustering approach follows bottom up approach. In this each data point is assigned as cluster and after computing each iteration the formed clusters are merged depending on the criteria chosen. The agglomerative clustering is ($n^2\log(n)$). The hierarchical-based clustering has a diverse benefit that the distance can be computed by using any valid measure. In point, the opinion themselves are not essential rather all computation is done by using matrix of distances like Euclidean which is a distant matrix used.

Spectral clustering algorithm is an algorithm in which data points are portioned by means of similarity matrix (Malkov and Yashunin, 2018). This works in three stages, i.e., pre-processing which focus on building of similarity matrix, construction of eigen vectors which is done by spectral mapping, post processing which deals with grouping data points. Spectral clustering algorithm is easy to analyse empirically, and it works very faster, it exhibits very high computational complexity.

Grid-based clustering algorithm is an algorithm in which operations are done on grids and that grids are formed by the objects space. The major advantage of this algorithm is that this does not need the computation of distance and further the clustering is done based on the obtained summarised data points, the complexity of this algorithm is ($O$).

Density-based clustering algorithm in an algorithm in which a cluster is continuously growing till the density in the region surpasses the threshold. The density algorithm is valuable in dealing with the noise available in a dataset. It requires only a single scan of the input datasets and parameters associated with density which are to be initialised. A partitional clustering algorithm partitions the data points into k partition where each partition represents a cluster. It has two properties each group should contain an object and each object should belong to one group. *k*-means clustering comes under portioning clustering. So many algorithms came till now for *k*-means like for numeric data, categorical data, and mixed data (Lana et al., 2015). In generally *k*-means clustering data in divided into different clustering by selecting centroids for clusters. Initially algorithm takes two inputs. The dataset containing n objects and *k*-number of clusters that are going to be created. Firstly, the centroids are selected randomly and then data points are assigned to clusters by measuring Euclidean distance. There are some other distant metrics that can be used are Manhattan. When all of the data points are assigned to some clusters then first iteration is completed and an early grouping is done but this clustering is rejected because it is applicable only for numeric data and it can be applied for categorical data and it is computationally expensive time complexity being (*nKI*), where *n* is the no. of data points, be the clusters that are going to form, and *I* are the no. of iterations). In the proposed algorithm, an approach to systematically selecting the initial centroids has been proposed. In this initially the given data points are plotted in 2D. All the data points should have positive values and if negative value then first is converted into positive value this is necessary because distance is calculated from the origin.

Elavarasi et al. (2011) constructs *k*-partition of all the data points and each obtained partition signifies a cluster. It has two properties each group should contain an object and each object should belong to one group. Partitioning clustering is also known as non-hierarchical clustering as every instance is positioned in precisely one of *k*-commonly exclusive clusters. In this clustering, the user needs to input the preferred count of clusters *k*-as only a single set of cluster is the outcome of a typical partitioned clustering algorithm. As discussed, in this clustering the user needs to give the count of clusters (*k*-) and from computation point of view the algorithm initiates the centres also called centroids of the *k*-partitions. In conclusion, *k*-means then allocates the data points based on the existing centres and re-compute centres based on the other available data points. These steps are recurrent until a definite objective function is optimised based on intracluster similarity or inter-cluster dissimilarity. Therefore, functional initialisation of centroids is a very significant factor for getting the effective and efficient results from partitional approach.

This paper focusses on *k*-means clustering which can be applied on data like numeric, categorical, and mixed data (Lim et. al., 2012). In general, *k*-means clustering data in divided into different clustering by selecting centroids for clusters. In the proposed algorithm, the numbers of clusters are chosen by using the heuristic as depending upon the set of similar datasets. Then the centroid is selected using the defined objective function and further all the iterations are performed.

## 2 *k*-means clustering

In general, the *k*-means clustering needs two inputs the dataset and the count of clusters required. This helps us to classify *n* number of data points into *k*-clusters. Similarity of clusters is known by measuring the Euclidean distance between the objects. Various distant metrics can also be used along with Euclidean distance like Manhattan distant metric, Minkowski distance metric, Mahalanobis metric, etc. Now, take the mean value of clusters as centre of gravity. Firstly, the centroids are selected randomly as the centre of cluster and every data point is allocated to a given cluster by computing the Euclidean distance for considering the computational efficiency. The first iteration consists of the task of assigning all the available data points to the relevant clusters formed. Then algorithm starts new iteration and then again, we find the new centroids and finally, a situation will come when the algorithm will attain its objective function.

The *K*-means algorithm takes two inputs the *n* no. of objects and *k* – no. of clusters. This helps us to classify *n* no. of data points into *k*-clusters. Similarity of clusters is known by measuring the Euclidean distance between the objects. Various distant metrics can also be used along with Euclidean distance like Manhattan distant metric, Minkowski distance metric. Now, take the mean value of clusters as centre of gravity. Firstly, the centroids are selected randomly as the centre of cluster and each data point is assigned to given cluster by measuring the Euclidean distance. Another distant metric can also be used. When all the data points are assigned to some clusters then first iteration is done. Then algorithm starts new iteration and then again, we find the new centroids and finally, a situation will come when the centroids do not move anymore, or the data objects do not change their cluster. This shows the convergence criterion for clustering. Algorithm for *k*-means is given in Algorithm given below:

Algorithm for k-means is as follows:

```
Input: $v = \{v_1, v_2, v_3, \dots, v_n\}$
Output. $k$ = the number of desired clusters
Method:
Select centroids known as initial centroid
Assign each data point to cluster by calculating Euclidean distance by Euclidean distance formula
Compute mean formula till the convergence is met
```

This algorithm is easy to implement on large datasets, but it has some limitations too. This algorithm is applicable only for numeric data only this cannot be applied for categorical data. But this is computationally expensive, time complexity is O($nKl$), where is $l$ is no. of iterations, $n$ are the no. of data points and $k$ are the clusters that are going to be formed.

The final clusters depend on the selection of initial centroids. The result may be different for multiple runs of algorithm for the same input data.

## *2.1 k-means using Euclidean*

```
Input: $d = d1, d2, d3, \dots, dn$
Output: $k = n(C_i)$, where $C_i$ are the preferred number of clusters
Method:
    i.   Select $i_c$ as the initial centroid.
    ii.  ¥ $d_i$ Compute $dis$ as:
             $dis((x, y), (a, b)) = \sqrt{(x-a)^2 + (y-b)^2}$
    iii. Next, compute $\mu$ ¥ ($d_i$)
```

The above $k$ means clustering is done by using the Euclidean Distance and as discussed, we can use multiple metric for computing the distance *k*-means like Manhattan, Minkowski, etc. (Sinwar and Kaushik, 2014).

## *2.2 k-means using Manhattan distance metric*

```
Input set of $d_i$ and $C_i$.
$di = d1, d2, d3, \dots, dn$ // data points
And //clusters
```

```
i.   ¥ $d_i$ $U$ selected $c_i$ Compute $dis$ as:
     $dis(x, y) = |x_1 - x_2| + |y_1 - y_2|$
ii.  Compute cn as centre using the formula
     $\left(\frac{1}{ci}\right)\sum_{1}^{c1} x1$
iii. Re-compute dis(cn, di)
iv.  Repeat until (¥ data points) ← cluster ($c_i$)
```

### 2.3 k-means using Minkowski distance metric

```
Input set of $d_i$ and $C_i$.
$d = d1, d2, d3, \dots, dn$ // data points
And $ci = c1, c2, c3, \dots, cn$ //clusters
    i.   ¥ $d_i$ -> rand(di) as $C_i$ centre
    ii.  ¥ data ¥ $d_i$ U selected ci compute dis as:
         $dis(xy) = |x_{ik} - x_{jk}|$
    i.   Compute cn as center using the formula
         $\left(\frac{1}{ci}\right)\sum_{1}^{c1} x1$
    iii. Ci <- di € min(dis(xy)) to the cluster
    iv.  Re-compute dis(cn, di)
```

In conclusion, by analysing the results of different distance metric, it is noted that *k*-means is done using the Euclidean distance because it gives the most efficient result and moreover, it is space oriented. Result for *k*-means using Manhattan and Euclidean is almost same, it is just that Manhattan gives the more distortion (Singh et al., 2013). In next sub section we will discuss the algorithm related for two-dimensional datasets.

## 3 Algorithm for 2D dataset

In heuristic approach-based clustering algorithm, two-dimensional dataset is taken as input. This dataset is based on the numeric form having both positive and negative values. If the datasets containing the negative values, then, firstly that negative value is converted to positive value to lay all data points on the identical plane. The minimum value for $x$ axis will $x_{\min}$ be and minimum value for $y$ axis will be $y_{\min}$. Then, all the data points from the datasets are subtracted from the minimum values. Next, all the positive value data points from the boundary of rectangle which is divided into $k$ clusters. After selecting the centre data point distance of each data point is computed with respect to each centroid. The heuristic function designed in this paper is based on the conceptually similar data points so that they are grouped into accurate clusters. The heuristic function is computed by finding the *minimum*($x$), *maximum*($y$), *minimum*($y$) and *maximum*($y$). The algorithm for 2D datasets is as follows:

```
   Input: $P = \{p_1, p_2, p_3, \ldots, p_n\}$ where P are 2D points.
   Output: $M = \{M_1, M_2, \ldots, M_n\}$ where M are the formed clusters.
1  If € (+ve, –ve) data points in input dataset then go to step 2 else go to step 3
2  ¥ × ¥ coordinate, compute $x_{minimum}$, and $y_{minimum}$
3  Sub(¥di) from Min($x_{minimum}$, $y_{minimum}$) from step 2
4  Construct the rectangle boundary values:
   Compute minimum(x), maximum(y), minimum(y), maximum(y)
5  Next, construct cluster Ci from the rectangle by using the heuristic value k as
   μ(minimum(x), maximum(y), minimum(y), maximum(y)).
6  ¥ each cluster ← dis (϶ data point).
7  Do ¥ data point computed di»centroid cjΠ each cluster j ← assign data points. //for each
   cluster data points are assigned
8  Set cluster d[i] = j
9  Set Ʋ Dist[i] = d(di ∪ cj).
10 ¥ each cluster $d_i$ (i <= j <= k), re-compute the centroids.
11 If dis <= present nd,then pi ϶ sc //dis is the distance computed
12 Else
13 ¥ centroid cj compute the dis(di ∪ cj) //distance computed between di and ci
14 Repeat until the objective function is met.
```

## 3.1 Illustrative example of 2D algorithm

Here, we are going to explain 2D algorithm using an example. In Table 1 we have taken a sample dataset with 15 points and showing their *x*-axis and *y*-axis, respectively.

The minimum and maximum from this new table:

- $x_{\min} = 0, x_{\max} = 14$
- $y_{\min} = 0, y_{\max} = 16$
- boundary values (14, 0) and (0, 16).

Now form a rectangle and divide it in to four parts with four centroids ($R_1$, $R_2$, $R_3$, $R_4$):

- $R_1$ = (4.5, 13), $R_2$ = (11.5, 13), $R_3$ =( 4.5, 5), $R_4$ = (11.5, 5).

Following are the iterations as:

1. Iteration 1
   - $R_1$ -> $D_4$, $D_2$, $D_{11}$, $R_2$ -> $D_3$, $D_1$, $D_7$, $D_{14}$, $D_{10}$, $R_3$ -> $D_9$, $D_5$, $D_{13}$, $D_{12}$, $R_4$ -> $D_8$, $D_6$, $D_{15}$.
2. Iteration 2
   - $R_1$ -> $D_{11}$, $D_2$, $D_4$, $R_2$ -> $D_1$, $D_3$, $D_7$, $D_{10}$, $D_{13}$, $R_3$ -> $D_6$, $D_9$, $D_{11}$, $D_{13}$, $R_4$ -> $D_7$, $D_9$, $D_{15}$.

| | |
|---|---|
| $R_1$ | $R_2$ |
| $R_3$ | $R_4$ |

**Table 1** Sample 2D dataset

| *Data points* | *X* | *Y* |
|---|---|---|
| $D_1$ | 6 | 2 |
| $D_2$ | 6 | –3 |
| $D_3$ | 4 | 3 |
| $D_4$ | 5 | –5 |
| $D_5$ | –6 | –4 |
| $D_6$ | –4 | 7 |
| $D_7$ | 7 | 6 |
| $D_8$ | 3 | –2 |
| $D_9$ | –4 | –2 |
| $D_{10}$ | 4 | 6 |
| $D_{11}$ | –6 | 8 |
| $D_{12}$ | –5 | –8 |
| $D_{13}$ | –8 | –3 |
| $D_{14}$ | 1 | 6 |
| $D_{15}$ | 1 | –6 |

**Table 2** Computed values

| *Data points* | *X* | *Y* |
|---|---|---|
| $D_1$ | 11 | 14 |
| $D_2$ | 6 | 14 |
| $D_3$ | 12 | 13 |
| $D_4$ | 5 | 13 |
| $D_5$ | 6 | 4 |
| $D_6$ | 15 | 5 |
| $D_7$ | 14 | 16 |
| $D_8$ | 12 | 7 |
| $D_9$ | 5 | 7 |
| $D_{10}$ | 12 | 14 |
| $D_{11}$ | 3 | 17 |
| $D_{12}$ | 4 | 1 |
| $D_{13}$ | 1 | 5 |
| $D_{14}$ | 9 | 15 |
| $D_{15}$ | 9 | 3 |

Next, in Table 3 we provide the comparative analysis of *k*-means, algorithm for the sample dataset taken in Table 1. In Table 3, we give the number of iterations respective to each of the algorithm. Also, in Table 4, the comparative analysis of iris dataset is given for further empirical efficiency of the proposed algorithm.

**Table 3** Comparative analysis of sample dataset

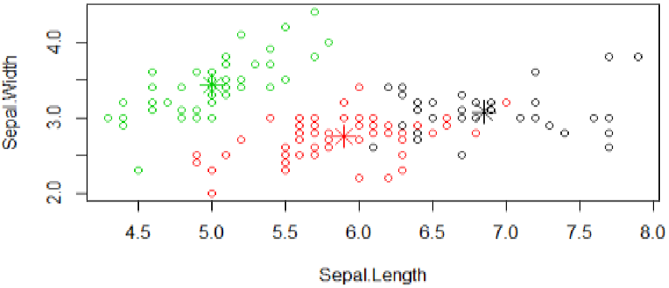
| *Clustering algorithm* | *Count of iterations* |
|---|---|
| K-means | 6 |
| Modified 2D algorithm for K-means | 4 |

**Table 4** Comparative analysis of iris dataset

| *Clustering algorithm* | *Count of iterations* |
|---|---|
| *k*-means | 73 |
| Modified 2D algorithm for *k*-means | 67 |

In conclusion, it is noted that 2D algorithm used for clustering is more efficiently computed to select initial centroid for each of the constructed cluster. The *k*-means also stemmed into precise output but it is complex in terms of space and time. In comparison to this existing, our proposed algorithm the initial centroids are not considered randomly but computed so that we move to right direction. The cluster formation of without heuristic *k*-means on the sample dataset is given in Figure 1.

**Figure 1** *k*-means (see online version for colours)



## 4 Related work

Güngör and Ünler (2007) has developed a new algorithm using simulated annealing *K*-harmonic means clustering (SAKHMC). The authors have done the empirical analysis on the iris dataset to show the efficiency of SAKHMC.

Elavarasi et al. (2011) proposed the review on the partition clustering algorithms. In this paper the authors describe the operational performance, the procedures to be monitored and the restrictions which affects the enactment of the algorithm. Clustering is majorly used tools in the domain of knowledge discovery. The authors also discussed the various different types of clustering algorithms.

Arockiam et al. (2012) have given the basic concepts of clustering. There are so many methods for clustering algorithms like hierarchical method which is further divided in to two parts agglomerative algorithms and divisive algorithms; partitioning methods which is further divided into four parts relocation, probabilistic, *k*-means, density-based which is further divided into two parts connectivity and functions clustering. *k*-means is part of partitioning clustering which partitions a dataset into a cluster.

Vij and Kumar (2012) has proposed a 2D algorithm in which centroids is not selected randomly. It was basically improved *k*-means algorithm in which initial centroids are selected using the authors proposed algorithm. When the datasets containing the negative values, then firstly that negative value is converted to positive. Next, the minimum values are computed for all *x* axis and *y* axis. So, this will make all the data points have positive value now these values form the boundary of rectangle which is divided into *k* clusters. After selecting the centroids distance of each centroid is computed in comparison to each centroid.

Islam and Chetty (2013) has given a novel approach for data clustering including various features like random generation of population, heuristic function, framework for knowledge representation, protein substructure of dataset considered for empirical analysis.

Mikaeil et al. (2018) has proposed the clustering algorithm using fuzzy c-mean and implemented the proposed approach using MATLAB. The authors determined the efficiency of the proposed algorithm by finding the hourly production rate of each studied dimension (Plank, 2018).

Various authors have given different approaches for clustering. Although they have covered many applications but there are some issues that are still need to be challenged. In our proposed algorithm we will give an algorithm for clustering the dataset.

## 5 Proposed algorithm

In this section we will discuss an approach for dividing the 2D data points based on the Manhattan distant metric and systematic selection of centroids using density which is created by using the heuristic function. As discussed earlier, to divide 2D clusters we have to convert all data points to some positive value but in proposed algorithm there is no need to take the positive values. We are using Manhattan distance because it gives the low distortion as compared to Euclidean distance.

Heuristic-based efficient 2D algorithm is given as follows:

```
      Input: $D = \{d_1, d_2, \ldots, d_n\}$
      Output: $C = \{C_1, C_2, \ldots, C_n\}$
      Steps:
i.    Find the density denoted by a symbol δ of each data point as
      $\delta = \frac{X \min i}{(Y \min i + 0.01)} * \alpha$
ii.   Data points with max(δ)¥D are selected as centroids
iii.  Final centroid Ci <- cosine similarity (max (δ), f(n)) where f(n) is the most significant
      cluster as per human analysis.
iv.   Compute dis э( data point and cluster centroid) as
      $dis(xy) = |x_1 - x_2| + |y_1 - y_2|$
v.    cluster $c_i$ <- with min(dis(xy))
vi.   do ¥ any cluster $c_n$ <- di
```

Illustrative example:

| *Points* | *X* | *Y* |
|---|---|---|
| P1 | 3 | 6 |
| P2 | –4 | 6 |
| P3 | 4 | 5 |
| P4 | –5 | 5 |
| P5 | –4 | –6 |
| P6 | 7 | –3 |

Step 1 Density of each data point by the formula given above:

$$P_1 = 0.59, P_2 = 2.98, P_3 = 0.59, P_4 = 0.1, P_5 = 4.3$$

Step 2 Points with highest density are chosen as centroids using the formula:

$$\delta = \frac{X \min i}{(Y \min i + 0.01)} * \alpha$$

where $\alpha$ is the constant with value 0.45 and $\delta$ is the density, i.e., $p_5$, $p_6$.

Step 3 Compute *dis*(*xmin*, *ymin*) for each data point with respect to centroid:

$$dis(x \min, y \min) = |x_1 - x_2| + |y_1 - y_2|$$

Distances from $p_5$:

$$P_1 = 16, P_2 = 11, P_3 = 16, P_4 = 17$$

Distances from $P_6$:

$$P_1 = 14, P_2 = 19, P_3 = 10, P_4 = 19$$

Step 4 Now data points will fall into cluster to which they are closer:

- $P_2$, $P_4$ will fall to cluster $P_5$
- $P_1$, $P_3$ will fall into cluster $P_6$.

First iteration will get completed here.

Step 5 Same points will get repeated now in the clusters $P_5$ and $P_6$.

The proposed algorithm gives the better centroid selection by using the computed heuristic function, which further helps in reducing the iteration of the clustering as compared to *k*-means clustering. However, the limitation in computation of heuristic function exists as the heuristic function is not tractable for some of the dataset having randomisation in the elements. The efficiency of the proposed algorithm in such dataset becomes complex in view of computation of heuristic function for which the sampling of the dataset requires time. In such cases the time complexity of the proposed algorithm increases by cn as compared to *O*(*n*log*n*).

## 6 Conclusions

In this paper we have given the review for clustering tools and techniques and also we proposed novel algorithm for *K-means* clustering for 2D dataset. We have divided the data based on the density to make it more efficient as compared to the other existing algorithms. We have found that after using the density concept there is no need to differentiate the positive and the negative data points and also Manhattan distance metric gives low distortion value as compared to Euclidean. The major drawback for *K-means* algorithm for 2D dataset was to firstly differentiate the positive and negative points then start clustering. In our proposed work there is no need to differentiate between the positive and negative points as the initial centroids are chosen according to the density of the data points.

## References

Arockiam, L., Baskar, S.S. and Jeyasimman, L. (2012) 'Clustering techniques in data mining', *Asian Journal of information Technology*, Vol. 11, No. 1, pp.40–44.

Elavarasi, S.A., Akilandeswari, J. and Sathiyabhama, B. (2011) 'A survey on partition clustering algorithms', *International Journal of Enterprise Computing and Business Systems*, Vol. 1, No. 1, pp.1–14.

Güngör, Z. and Ünler, A. (2007) 'K-harmonic means data clustering with simulated annealing heuristic', *Applied Mathematics and Computation*, Vol. 184, No. 2, pp.199–209.

Islam, M.K. and Chetty, M. (2013) 'Clustered memetic algorithm with local heuristics for ab initio protein structure prediction', *IEEE Transactions on Evolutionary Computation*, Vol. 17, No. 4, pp.558–576.

Lan, X., Li, Q. and Zheng, Y. (2015) 'Density K-means: a new algorithm for centers initialization for K-means', in *Software Engineering and Service Science* (*ICSESS*), *2015 6th IEEE International Conference*, IEEE, September, pp.958–961.

Lim, J., Jun, J., Kim, S.H. and McLeod, D. (2012) 'A framework for clustering mixed attribute type datasets', *Proc. of the 4th Int. Con. on Emerging Databases* (*EDB 2012*), Seoul, Korea.

Malkov, Y.A. and Yashunin, D.A. (2018) 'Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp.1–13

Maulik, U. and Bandyopadhyay, S. (2002) 'Performance evaluation of some clustering algorithms and validity indices', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 12, pp.1650–1654.

Mikaeil, R., Haghshenas, S.S., Haghshenas, S.S. and Ataei, M. (2018) 'Performance prediction of circular saw machine using imperialist competitive algorithm and fuzzy clustering technique', *Neural Computing and Applications*, Vol. 29, No. 6, pp.283–292.

Plank, P. (2018) 'Introduction', in *Price and Product-Mix Decisions under Different Cost Systems*, pp.1–5, Springer Gabler, Wiesbaden.

Singh, A., Yadav, A. and Rana, A. (2013) 'K-means with three different distance metrics', *International Journal of Computer Applications*, Vol. 67, No. 10, pp.13–17.

Sinwar, D. and Kaushik, R. (2014) 'Study of Euclidean and Manhattan distance metrics using simple K-means clustering', *International Journal for Research in Applied Science and Engineering Technology* (*IJRASET*), Vol. 2, No. 5.

Vij, R. and Kumar, S. (2012) 'Improved k-means clustering algorithm for two dimensional data', in *Proceedings of the Second International Conference on Computational Science*, *Engineering and Information Technology*, ACM, October, pp.665–670.