Empirical Evaluation of Distance Measures for Nearest Point with Indexing Ratio Clustering Algorithm

Raneem Qaddoura1, Hossam Faris2, Ibrahim Aljarah2, J. J. Merelo3 and Pedro A. Castillo3

1Information Technology, Philadelphia University, Amman, Jordan
2Department of Business Information Technology, King Abdullah II School for Information Technology, The University of Jordan, Amman, Jordan
3ETSIIT-CITIC, University of Granada, Granada, Spain

Keywords: Clustering, Cluster Analysis, Distance Measure, Nearest Point with Indexing Ratio, NPIR, Nearest Point, Indexing Ratio, Nearest Neighbor Search Technique.

Abstract: Selecting the proper distance measure is very challenging for most clustering algorithms. Some common distance measures include Manhattan (City-block), Euclidean, Minkowski, and Chebyshev. The so called Nearest Point with Indexing Ratio (NPIR) is a recent clustering algorithm, which tries to overcome the limitations of other algorithms by identifying arbitrary shapes of clusters, non-spherical distribution of points, and shapes with different densities. It does so by iteratively utilizing the nearest neighbors search technique to find different clusters. The current implementation of the algorithm considers the Euclidean distance measure, which is used for the experiments presented in the original paper of the algorithm. In this paper, the impact of the four common distance measures on NPIR clustering algorithm is investigated. The performance of NPIR algorithm in accordance to purity and entropy measures is investigated on nine data sets. The comparative study demonstrates that the NPIR generates better results when Manhattan distance measure is used compared to the other distance measures for the studied high dimensional data sets in terms of purity and entropy.

1 INTRODUCTION

Clustering is the task of grouping similar points to the same cluster and dissimilar points to different clusters (Han et al., 2011). It is used in many applications such as image processing (Kumar et al., 2018; Santos et al., 2017), dental radiography segmentation (Qaddoura et al., 2020a), pattern recognition (Liu et al., 2017; Silva et al., 2017), document categorization (Mei et al., 2017; Brodić et al., 2017), and financial risk analysis (Kou et al., 2014).

A recent clustering algorithm was proposed by (Qaddoura et al., 2020b), which is named Nearest Point with Indexing ratio (NPIR). NPIR uses the nearest neighbor search technique along with three novel operations, which are Election, Selection, and Assignment. It is a combination of partitional clustering and density-based clustering approaches as it uses iterations same as K-means, and also uses nearest neighbors search technique to find dense predicted clusters.

The original paper of NPIR (Qaddoura et al., 2020b) uses the Euclidean distance measure to find the nearest neighbors of a certain point. Authors of the algorithm argued that the performance of the algorithm decreases for high dimensional data sets due to the use of the Euclidean distance. Thus, in this work, we experiment NPIR using other distance measures such as Manhattan, Chebyshev, and Minkowski distance measures. The results of running the algorithm are evaluated using two well-known measures, which are purity and entropy.

The remainder of the paper is organized as follows: Section 2 presents recent work on clustering. Section 3 describes in brief the nearest neighbors search technique, the distance measures, and the NPIR algorithm. Section 4 discusses the experimental results. Finally, Section 5 concludes the work.

2 RELATED WORK

Clustering algorithms can be categorized into partitioning algorithms, hierarchical algorithms, and
density-based algorithms (Han et al., 2011; Chen et al., 2016; Lu et al., 2018). Partitioning algorithms include $K$-means (Jain, 2010), $K$-means++ (Arthur and Vassilvitskii, 2007), and Expectation Maximization (EM) (Han et al., 2011). Density-based algorithms include DBSCAN (Ester et al., 1996) and OPTICS (Ankerst et al., 1999). Hierarchical algorithms include BIRCH (Zhang et al., 1996) and HDBSCAN (Campello et al., 2015). Partitioning algorithms are not suitable for non-spherical clusters and they fall in local optima (Chen et al., 2016). Hierarchical algorithms are not suitable for clusters that are not well separated and they take more time and space compared to the partitioning algorithms (Lu et al., 2018). Density-based algorithms fail to detect clusters of different densities (Chen et al., 2016; Lu et al., 2018) and parameters tuning is very difficult for these algorithms. Another class of algorithms combining nature-inspired algorithms and partitional clustering algorithms can also be observed for clustering (Qaddoura et al., 2020c).

Due to the limitations of the aforementioned algorithms, work can still be found for solving the clustering task. A cluster weights are determined for each cluster in the MinMax $K$-means algorithm proposed by (Tzortzis and Likas, 2014). The work of (Frandsen et al., 2015) uses iterative $K$-means to cluster site rates by selecting the partitional schemes automatically, entropy-based farthest neighbor technique is used to find the initial centroids in $K$-means in the work of (Trivedi and Kanungo, 2017). A residual error-based density peak clustering algorithm (REDPC), is proposed in the work of (Parmar et al., 2019) to handle data sets of various data distribution patterns. An improved density algorithm named as DPC-LG, uses logistic distribution that is proposed in the work of (Jiang et al., 2019). The work of (Cheng et al., 2019) presents a Hierarchical Clustering algorithm Based on Noise Removal (HCBNR), which recognizes noise points and finds arbitrary shaped clusters. An ant-based method that takes advantage of the cooperative self-organization of Ant Colony Systems to create a naturally inspired clustering and pattern recognition method is proposed in (Fernandes et al., 2008). The work of (Stefatos and Siripoulos, 2004) considers a cluster-based combinatorial forecasting schemes based on clustering algorithms and neural networks with an emphasis placed on the formulation of the problem for better forecasts. Authors in (Pal et al., 1996) criticized the sequential competitive learning algorithms that are curious hybrids of algorithms used to optimize the fuzzy c-means (FCM) and learning vector quantization (LVQ) models by showing that they do not optimize the FCM functional and the gradient descent conditions are not the necessary conditions for optimization of a sequential FCM functional.

The work of (Bhattacharyya et al., 2016) discusses data sets with multiple dimensions. It shows how data clustering is applied on such data sets. Large and high dimensional data sets are experimented using a variant of the EM algorithm in the work of (Kadir et al., 2014). Parallel implementation of the Best of both Worlds (BoW) method on a very large moderate-to-high dimensional data set can be found in the work of (Ferreira Cordeiro et al., 2011). $GARDEN_{HD}$ is an effective, efficient, and scalable algorithm which is introduced in the work of (Orlandic et al., 2005) on multi-dimensional data set. In addition, several works can be found for large scale data sets in the work of (Al-Madi et al., 2014; Aljarah and Ludwig, 2012; Aljarah and Ludwig, 2013a; Cui et al., 2014).

Distance measures effect on clustering is analyzed and compared in several studies. The paper (Finch, 2005) presents a comparison of four distance measures in clustering with dichotomous data and their performance in correctly classifying the individuals. Other comparison can be found in the work of (Huang, 2008) by analyzing the effectiveness of five distance measures in partitional clustering for text document datasets. Further, a technical framework is proposed in the work of (Shirkhorshidi et al., 2015) to compare the effect of different distance measures on fifteen datasets which are classified as low and high-dimensional categories to study the performance of each measure against each category. The work of (Pandit et al., 2011) presents the effect of different distance measures based on application domain, efficiency, benefits and drawbacks. Other works can also be found in the literature which are specialized to specific domain or algorithm. Authors of (Klawonn and Keller, 1999) proposed a modified distance function of the fuzzy c-means based on the dot product to detect different cluster shape, lines, and hyper-planes. A study of the effect of different distance measures in detecting outliers using clustering-based algorithm for circular regression model is presented in (Di and Satari, 2017). Authors of (Paukkeri et al., 2011) studied the effect of dimensionality reduction on different distance measures in document clustering.

Due to the shortness of the partitional and the density-based algorithms on clusters of non-spherical shapes and clusters of different densities, respectively, a very recent algorithm was introduced by the authors of (Qaddoura et al., 2020b), which is named Nearest Point with Indexing Ratio (NPIR). It combines both behaviors of partitional and density-based algorithms by using iterations and the nearest neighbor search technique. It detects arbitrary shaped clusters having...
non-spherical clusters and clusters with different sizes or densities. However, it uses the Euclidean distance to calculate the distance between any two points. In some cases, this leads to low quality of results for high dimensional data sets. Thus, this work investigates other distance measures for calculating the distance between the points in the NPIR algorithm, to find the most appropriate distance measure for the high dimensional data sets.

3 GENERAL DEFINITIONS AND TERMINOLOGY

This section discusses the nearest neighbors search problem and the distance measures for finding the nearest neighbors. This section also discusses the NPIR algorithm, which is used in the experiments.

3.1 Nearest Neighbors and Distance Measures

Searching for the nearest neighbors of a point is used as part of performing the clustering task for many recent algorithms (Lu et al., 2018). It is used to cluster the closest points to a certain point to the cluster of that point. The nearest neighbor search problem can be defined as follows (Lu et al., 2018; Hoffmann, 2010):

Definition 1. Given a set of N points \( P = \{p_1, p_2, \ldots, p_N\} \) in space, find k-NN\((p_i) = \{nn_1, nn_2, \ldots, nn_k\}\) which represents the k-nearest neighbors set of a certain point \( p_i \) where \( p_i \in \{p_1, p_2, \ldots, p_N\} \), \( k = \text{k-NN}(p_i) \), and \( k < N \).

A distance measure is used to discover a nearest neighbor \( nn_j \in \{nn_1, nn_2, \ldots, nn_k\} \) to a point \( p_i \in \{p_1, p_2, \ldots, p_N\} \). Several distance measures can be used to find the nearest neighbor, which include Manhattan (Black, 2006), Euclidean (Anton, 2013), Chebyshev (Cantrell, 2000), and Minkowski (Grabusts et al., 2011).

The Minkowski distance measure generalizes other distance measures such as Euclidean and Manhattan distance measures having different values of \( r \) to calculate the distance between the point \( p_i \) and \( nn_j \). Minkowski distance can be defined as follows (Grabusts et al., 2011):

\[
Minkowski(p_i, nn_j) = \left( \sum_{f=1}^{d} |p_{if} - nn_{jf}|^r \right)^{1/r}
\]  

where \( d \) is the dimension or the number of features and \( f \) is the feature number. Manhattan, Euclidean, and Chebyshev distance measures are derived from the Minkowski distance measure where a \( p \) value of 1, 2, and \( \infty \) is determined, respectively. These measures can be defined by Equations 2, 3, and 4:

\[
\text{Manhattan}(p_i, nn_j) = \sum_{f=1}^{d} |p_{if} - nn_{jf}|
\]  

\[
\text{Euclidean}(p_i, nn_j) = \sqrt{\sum_{f=1}^{d} (p_{if} - nn_{jf})^2}
\]  

\[
\text{Chebyshev}(p_i, nn_j) = \lim_{r \to \infty} \left( \sum_{f=1}^{d} |p_{if} - nn_{jf}|^r \right)^{1/r}
\]  

\[
= \max_f |p_{if} - nn_{jf}|
\]  

3.2 Nearest Point with Indexing Ratio (NPIR) Algorithm

Nearest Point with Indexing Ratio (NPIR) is a recent clustering algorithm with three parameters: the number of clusters \( (k) \), the indexing ratio \((IR)\), and the number of iterations \((i)\). NPIR consists of three main operations, namely the Election, the Selection, and the Assignment. The Election simply selects an assigned point in space and names it as Elected. The Selection considers selecting the k-NN point to the Elected point and naming it as Nearest. The Assignment considers assigning the Nearest to the cluster of the Elected, and marking the Elected as the Assigner, if the Nearest is not assigned or the Nearest is closer to the Elected than its original Assigner. At each assignment/reassignment of the Nearest to the cluster of the Elected, the descendants of the Nearest are clustered to the cluster of the Elected. K-dimensional tree (Maneeewongvatana and Mount, 1999) is used to find the Nearest point to an Elected point. The Euclidean distance is used to find the distance between the points\(^1\).

The algorithm uses an iterative process to enhance the quality of the clustering results. At each iteration, multiple Election, Selection, and Assignment operations are performed, which are controlled by the IR parameter. That is, multiple considerations for selecting a Nearest point for an Elected point and considerations of assigning/reassigning the Nearest to the cluster of the Elected. Pseudocode 1 represents the steps of the algorithm. Lines 2 – 5 represents the initial steps of the algorithm. Lines 6 – 24 represents

\(^1\)http://evo-ml.com/npir/
Algorithm 1: NPIR Pseudo Code (Qaddoura et al., 2020b).

Input: Points, K, IR, i
Output: The predicted assignments
1: procedure NPIR
2: Initialize index, iterations
3: Create K-dimensional tree for the points
4: TotalIndex \(=\) Round\((IR \times (#Points)^2)\)
5: Select \(k\) random points as the initial points for the clusters
6: repeat
7: \hspace{1em} repeat
8: \hspace{2em} Randomly elect an assigned point and mark it as \textit{Elected} \((E)\)
9: \hspace{2em} Select the \(k\)-NN point of the \textit{Elected} \((N)\)
10: \hspace{2em} Increment the \(k\) value of the \(k\)-NN for the \textit{Elected} by 1
11: \hspace{2em} if \textit{Nearest} is not assigned yet to a cluster or \textit{Nearest} is assigned to a different cluster than the \textit{Elected} and distance\((N,E)\)<distance\((N,A)\) then
12: \hspace{2em} Assign the \textit{Nearest} and its descendants to the cluster of the \textit{Elected}
13: \hspace{2em} Mark the \textit{Elected} as the \textit{Assigner} \((A)\) for the \textit{Nearest}
14: \hspace{2em} Add the \textit{Nearest} as a descendant to the \textit{Assigner}
15: \hspace{2em} if Old cluster of the \textit{Nearest} becomes empty then
16: \hspace{2em} Assign a random point to the old empty cluster of the \textit{Nearest}
17: \hspace{2em} end if
18: \hspace{2em} end if
19: \hspace{1em} until All points are clustered and \(index++>\text{TotalIndex}\)
20: \hspace{1em} end if
21: \hspace{1em} until iterations++\(>i\)
22: \hspace{1em} Set the pointer to the first element of the distance vector for all points
23: \hspace{1em} return the predicted assignments
24: end procedure

the iterations performed where each iteration consists of multiple inner iterations displayed at lines 7 – 22. Inner iterations represents the multiple considerations of the \textit{Election} (line 8), \textit{Selection} (lines 9 – 11), and \textit{Assignment} (lines 12 – 21) operations.

4 EXPERIMENTAL RESULTS

This section presents the environment, evaluation measures, a presentation of the data sets, and the discussion of the results.

4.1 Environment

We ran the experiments on a personal computer with Intel core i7-5500U CPU @ 2.40GHz/8 GB RAM. For experiments, we used Python 3.7 and the Scikit Learn Python library (Pedregosa et al., 2011) to evaluate the algorithm using different distance measures.

4.2 Evaluation Measures

The results which are obtained from running the NPIR algorithm, are evaluated using the purity and entropy measures (Aljarah and Ludwig, 2013b). High purity and low entropy values indicate better clustering results (Qaddoura et al., 2020a).

Given \(L\) as the true labels of \(N\) points and \(R\) as the predicted clusters of these points. The purity and entropy measures can be formalize as follows (Aljarah and Ludwig, 2013b):

\[
Purity = \frac{1}{N} \sum_{j=1}^{k} \max_{i} (|L_{i} \cap R_{j}|) \tag{5}
\]

where \(R_{j}\) presents all points assigned to cluster \(j\), \(k\) is the number of clusters, and \(L_{i}\) is the labels of the points in cluster \(i\).

\[
Entropy = \sum_{j=1}^{k} \left( \frac{|R_{j}|}{n} E(R_{j}) \right) \tag{6}
\]

where \(E(R_{j})\) is the individual entropy of a cluster. Individual cluster entropy is calculated using Equation
\[ E(R_j) = -\frac{1}{\log k} \sum_{i=1}^{k} \frac{|R_j \cap L_i|}{R_j} \log \left( \frac{|R_j \cap L_i|}{R_j} \right) \] (7)

4.3 Data Sets

Data sets with different number of features are selected to evaluate the NPIR algorithm using different distance measures. The aim is to find a correlation between the number of features and the best distance measure, which suits the NPIR algorithm on the selected data sets. The data sets are gathered from UCI machine learning repository\(^2\) (Dheeru and Karra Taniskidou, 2017). Table 1 shows the name, number of clusters \((k)\), number of points \((n\text{instances})\), and number of features for each data set.

4.4 Results and Discussion

To evaluate NPIR algorithm using different distance measures, the experiments are performed for 30 independent runs for each data set. The average purity and entropy results are listed in Tables 2 and 3, respectively, for different distance measures having different \(p\) values for Minkowski, which are 1, 2, 4, 8, and \(\infty\). The \(p\) values of 1, 2, and \(\infty\) represent the Manhattan, Euclidean, and Chebyshev distance measures, respectively.

Table 2 shows that datasets with low dimensions including Iris 2D, Iris, Diagnosis II and Seeds data sets, having 2, 4, 6, and 7 features, respectively, have the best average results for different distance measures which are Euclidean, Manhattan/Euclidean, Chebyshev, and Chebyshev, respectively. In contrast, Manhattan distance measure is recommended to be used for high dimensional data sets as it shows the highest average values of purity for the remaining high dimensional data sets. This recommendation is consistent with the other studies in the literature (Pandit et al., 2011; Aggarwal et al., 2001; Aggarwal et al., 2019; Song et al., 2017; Tolentino and Gerardo, 2019) in which Manhattan distance measure has proven to give the best performance for high dimensional datasets for k-means (Aggarwal et al., 2001), Partial Least Square (PLS) and PLS discriminant analysis (PLS-DA) (Song et al., 2017), and Fuzzy C-Means (Tolentino and Gerardo, 2019) algorithms. It is also observed from Table 2 that Chebyshev has relatively bad results for high dimensional data sets compared to the other distance measures. Thus, Chebyshev is not recommended to be used for NPIR on high dimensional data sets. We can also observe that the same purity values are achieved by different distance measures for data sets having unbalanced data, which include Pop failure dataset and Unbalanced data set.

Table 3 generates similar observations given by Table 2. It shows that Iris 2D, Iris, Diagnosis II, and Seeds data sets, having low dimensions of values 2, 4, 6, and 7, respectively, have the best average results for different distance measures which are Euclidean, Manhattan/Euclidean, Chebyshev, and Chebyshev, respectively. In contrast, Manhattan distance measure is recommended to be used for high dimensional data sets and Chebyshev is not recommended for such data sets. In addition, and as observed from Table 2, unbalanced data set has the same values for different distance measures.

Since more features are considered for high dimensional datasets than lower dimensional ones, Manhattan distance gives better results as it calculates the distance between two points without exaggerating the discrepancy of the features, which is found in the other distance measures of this study. That is, when a point is close to another one for most features but not for few, Manhattan distance kind of shrug the few features off and is influenced by the distance of most features. This is not recognized for datasets with low dimensions since the possibility of having this discrepancy is minimal. In addition, NPIR algorithm considers iterative correction of wrongly clustered points with random selection of different Elected points, which makes it suitable to Manhattan distance for high dimensional datasets, since shrugging the few features off when calculating the distance can be corrected in later iterations if the effect of such few features is detected to be of more importance for a different Elected and corresponding Nearest.

Figures 1 and 2 also represent the results obtained from running the algorithm. The radar lines represent the purity and entropy values for each distance measure. The range of the values starts with the center of the radar at the worst value possible for the measure, and ends with the boundaries of the radar at the best value possible. This means that the range of the purity values starts at 0 and reaches 1 whereas the range of the entropy values starts at 1 and reaches 0. It is observed from the two figures that the radar line of the high dimensional data sets for the City block (Manhattan) distance measure is surrounding the other radar lines for the other measures. This indicates better clustering results having high values of purity and low values of entropy for the City block distance measure. In contrast, the radar line of the high dimensional data sets for the Chebyshev distance
Table 1: Data sets properties.

<table>
<thead>
<tr>
<th>Data set</th>
<th>No. of clusters(k)</th>
<th>No. of instances</th>
<th>No. of features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris 2D</td>
<td>3</td>
<td>150</td>
<td>2</td>
</tr>
<tr>
<td>Iris</td>
<td>3</td>
<td>150</td>
<td>4</td>
</tr>
<tr>
<td>Diagnosis II</td>
<td>2</td>
<td>120</td>
<td>6</td>
</tr>
<tr>
<td>Seeds</td>
<td>3</td>
<td>210</td>
<td>7</td>
</tr>
<tr>
<td>Zoo</td>
<td>7</td>
<td>101</td>
<td>16</td>
</tr>
<tr>
<td>Pop failures</td>
<td>2</td>
<td>540</td>
<td>18</td>
</tr>
<tr>
<td>Unbalanced</td>
<td>2</td>
<td>856</td>
<td>32</td>
</tr>
<tr>
<td>Soybean small</td>
<td>4</td>
<td>47</td>
<td>35</td>
</tr>
<tr>
<td>Divorce</td>
<td>2</td>
<td>170</td>
<td>54</td>
</tr>
</tbody>
</table>

Table 2: Purity results for applying Manhattan, Euclidean, Chebyshev, and Minkowski distance measures on NPIR algorithm.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Manhattan</th>
<th>Euclidean</th>
<th>Minkowski (p=4)</th>
<th>Minkowski (p=8)</th>
<th>Chebyshev</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris 2D</td>
<td>0.66</td>
<td>0.9</td>
<td>0.69</td>
<td>0.74</td>
<td>0.53</td>
</tr>
<tr>
<td>Iris</td>
<td>0.89</td>
<td>0.89</td>
<td>0.77</td>
<td>0.75</td>
<td>0.74</td>
</tr>
<tr>
<td>Diagnosis II</td>
<td>0.84</td>
<td>0.84</td>
<td>0.81</td>
<td>0.8</td>
<td>0.91</td>
</tr>
<tr>
<td>Seeds</td>
<td>0.7</td>
<td>0.66</td>
<td>0.7</td>
<td>0.65</td>
<td>0.74</td>
</tr>
<tr>
<td>Zoo</td>
<td>0.69</td>
<td>0.68</td>
<td>0.67</td>
<td>0.67</td>
<td>0.49</td>
</tr>
<tr>
<td>Pop failures</td>
<td>0.91</td>
<td>0.91</td>
<td>0.91</td>
<td>0.91</td>
<td>0.91</td>
</tr>
<tr>
<td>Unbalanced</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td>Soybean-small</td>
<td>0.97</td>
<td>0.92</td>
<td>0.9</td>
<td>0.93</td>
<td>0.37</td>
</tr>
<tr>
<td>Divorce</td>
<td>0.92</td>
<td>0.87</td>
<td>0.9</td>
<td>0.84</td>
<td>0.6</td>
</tr>
</tbody>
</table>

Table 3: Entropy results for applying Manhattan, Euclidean, Chebyshev, and Minkowski distance measures on NPIR algorithm.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Manhattan</th>
<th>Euclidean</th>
<th>Minkowski (p=4)</th>
<th>Minkowski (p=8)</th>
<th>Chebyshev</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris 2D</td>
<td>0.47</td>
<td>0.2</td>
<td>0.42</td>
<td>0.37</td>
<td>0.67</td>
</tr>
<tr>
<td>Iris</td>
<td>0.22</td>
<td>0.22</td>
<td>0.33</td>
<td>0.35</td>
<td>0.36</td>
</tr>
<tr>
<td>Diagnosis II</td>
<td>0.28</td>
<td>0.25</td>
<td>0.3</td>
<td>0.32</td>
<td>0.22</td>
</tr>
<tr>
<td>Seeds</td>
<td>0.54</td>
<td>0.6</td>
<td>0.53</td>
<td>0.6</td>
<td>0.5</td>
</tr>
<tr>
<td>Zoo</td>
<td>0.37</td>
<td>0.38</td>
<td>0.39</td>
<td>0.39</td>
<td>0.71</td>
</tr>
<tr>
<td>Pop failures</td>
<td>0.42</td>
<td>0.42</td>
<td>0.42</td>
<td>0.42</td>
<td>0.42</td>
</tr>
<tr>
<td>Unbalanced</td>
<td>0.11</td>
<td>0.11</td>
<td>0.11</td>
<td>0.11</td>
<td>0.11</td>
</tr>
<tr>
<td>Soybean-small</td>
<td>0.04</td>
<td>0.12</td>
<td>0.13</td>
<td>0.09</td>
<td>0.9</td>
</tr>
<tr>
<td>Divorce</td>
<td>0.24</td>
<td>0.33</td>
<td>0.29</td>
<td>0.4</td>
<td>0.86</td>
</tr>
</tbody>
</table>

measure indicates worst clustering results having low values of purity and high values of entropy. In addition, identical values of purity and entropy can be observed for Pop failure and Unbalanced data sets.

5 CONCLUSION

Identifying the right distance measure for an algorithm, which gives the best quality results for a selected dataset, is a good practice in clustering. NPIR is a recent algorithm which considers the distance between points to perform the clustering task. In this paper, an experimental study is done using different distance measures to find the impact of the distance measurement method on the performance of NPIR algorithm. In this paper, the distance is measured between the points in space and the corresponding nearest neighbors for the K-dimensional tree data structure, which is used in the NPIR algorithm. The results of the experiments show that Manhattan distance measure has the best average purity and entropy values for high dimensional data sets, whereas Chebyshev has the worst values for these data sets. The results also show that close and identical values of purity and entropy are achieved for low dimensional data sets and unbalanced data sets. Manhattan distance measure is best suited for NPIR for high dimensional datasets for two reasons: First, Manhattan distance measure calculates the distance between two points without exaggerating the discrepancy of the features. Second, the iterative correction of wrongly clustered...
Figure 1: Radar chart for the purity values for applying City-block (Manhattan), Euclidean, Minkowski (p=4), Minkowski (p=8), and Chebyshev distance measures on NPIR algorithm.

Figure 2: Radar chart for the entropy values for applying City-block (Manhattan), Euclidean, Minkowski (p=4), Minkowski (p=8), and Chebyshev distance measures on NPIR algorithm.

points in NPIR overcomes the possible shrugging of the few features off when calculating the Manhattan distance if the effect of such few features is detected to be of more importance for different Elections of points.

For future work, different evaluation measures can be investigated for measuring the performance of NPIR for different distance measures. We can use evaluation measures besides the purity and entropy measures, which might include Homogeneity Score, Completeness Score, V-Measure, Adjusted Rand Index, and Adjusted Mutual Information, extending the practical validity of the work. The effect of different distance measures can also be experimented on different algorithm than NPIR, which might include neural network or deep learning algorithms.
ACKNOWLEDGEMENTS

This work is supported by the Ministerio espa˜nol de Econom´ıa y Competitividad under project TIN2017-85727-C4-2-P (UGR-DeepBio).

REFERENCES


