REDCLAN - RELATIVE DENSITY BASED CLUSTERING AND ANOMALY DETECTION

Diptarka Saha[1], Debanjana Banerjee[2], Bodhisattwa Prasad Majumder[3]

[1,2]WalmartLabs, Bengaluru, Karnataka, India
[3]Dept. of Computer Science and Engineering,
University of California, San Diego, USA

ABSTRACT

Cluster analysis and Anomaly Detection are the primary methods for database mining. However, most of the data in today's world, generated from multifarious sources, don't adhere to the assumption of single or even known distribution - hence the problem of finding clusters in the data becomes arduous as clusters are of widely differing sizes, densities and shapes, along with the presence of noise and outliers. Thus, we propose a relative-KNN-kernel density-based clustering algorithm. The un-clustered (noise) points are further classified as anomaly or non-anomaly using a weighted rank-based anomaly detection method. This method works particularly well when the clusters are of varying variability and shape, in these cases our algorithm not only finds the "dense" clusters that other clustering algorithms find, it also finds low-density clusters that these approaches fail to identify. This more accurate clustering in turn helps reduce the noise points and makes the anomaly detection more accurate.

KEYWORDS

Clustering, Relative KNN – kernel density, Varying density clusters, Anomaly Detection, DBSCAN

1. INTRODUCTION

In the industry today, data categorization could be the single most important problem - categorize people according to income, categorize customers according to purchase patterns, categorize items according to price and the list goes on. The underlying data for categorization could have any form whatsoever – structured, unstructured, labelled, unlabelled, adhering to assumptions or, not.

In a scenario where the data points are unlabelled, the purpose of categorization would be simply to study the underlying pattern of the data and problems such as these are typically tackled under the realm of *clustering* [1]. In its core, clustering is an exploratory tool for data analysis and has been used in several fields such as statistics, document retrieval, image segmentation [2], biological sciences [3], psychology [4] etc. for some time now. In this paper, we are going to study a novel clustering technology that outperforms existing methods particularly well when the intrinsic clusters are of varying variability, shape& size. This more accurate clustering in turn helps reduce the noise points and makes the anomaly detection more accurate.

Natarajan Meghanathan et al. (Eds) : SOFE, ADCOM, ITCSS - 2018
pp. 25–39, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.81303

2. LITERATURE SURVEY

Cluster analysis is a challenging task and despite being studied by numerous authors for decades it still has some documented issues associated with it. This is primarily due to the vagueness of the success criterion of the problem – clustering attempts at separating unlabelled data into *meaningful* groups with minimal input from user; however, what defines meaningful is dependent on the application at hand. In the most general of meanings the objects are clustered or grouped based on the principle of *maximizing the intraclass similarity and minimizing the interclass similarity*[5]. The diverse application potential along with the heterogeneity of the data associated with these applications have given birth to several clustering techniques over the years. Below, we present a brief high-level review of these techniques.

Traditionally clustering techniques are roughly divided into *hierarchical* and *partitioning*[6] methods.

Hierarchical clustering defines a *dendrogram* representing nested groups of objects which represent a hierarchical relationship of these objects [7]. One can either start with a singleton and recursively merge clusters as deemed appropriate: this is known as *agglomerative* (bottom-up) technique. Reversely, one may start by putting all the points into one cluster and recursively split each cluster into daughter clusters until a stopping criterion is achieved, such methods are known as *divisive* (top-down). Example of hierarchical clusters are BIRCH[8], ROCK[9] etc. These algorithms are especially useful when data might have some inherent hierarchical structure and/or features in the data are of varying type. However, they typically tend to be of higher time complexity, also they suffer from the vagueness of the stopping criterion.

On the other hand, partitioning methods cluster the data directly and not in gradual steps. One kind of partitioning method typically attempt at iteratively finding certain centres of data as representative of a cluster and build cluster around those centres. K-means [10] and K-medoid [11] are renowned cluster techniques that use this method. Although these methods are more computationally efficient, they perform poorly when clusters are non-convex and/or data has outliers. The number of clusters is also user – defined which calls for domain knowledge.

Another kind of partitioning method is popularly known as *density-based clustering,* herehigh-density data spaces are identified and points in those spaces are put in the same cluster. DBSCAN is the most well-known density-based clustering technique[12]. OPTICS[13] is another such algorithm which attempts at alleviating shortcomings of DBSCAN. One major characteristic of these methods is the number of clusters to be found is not user – defined and hence it allows for more flexibility. Also, these algorithms typically don’t cluster all the points and may leave some points as noise points which may be treated as potential anomalies. Density based algorithms have proven to be improvement over other partitioning methods when data may have arbitrary shape and/or outliers; however, have historically failed to captured clusters of varying density – this drawback is precisely what we are hoping to eliminate while retaining all the aforementioned advantages.

Coming to Anomaly detection, it deals with the problem of finding points in data that do not conform to *expected* behaviour [14].As in the case of clustering defining what is *expected* or *normal* is the most difficult job and is typically application dependent. Most basic approaches [15][16] use the method of flagging off the most extreme points, points which typically fall beyond a certain threshold, these thresholds are mostly higher quantiles. Even though, the approach might be non-parametric, it fails to look at any more than what the data has to suggest at its surface. Another parametric alternative, Minimum Volume Ellipsoid estimation (MVE)[17] fits the smallest permissible ellipsoid volume around the majority of the data which represents *densely populated normal region*. Among other outlier detection techniques, *density-based anomaly detection* techniques such as Local Outlier Factor (LOF)[18] are also very popular due to their intuitive approache and interpretability. A comprehensive survey on outlier detection methods can be found in [19].

We however, instead of looking for anomalies overall, look for anomalies with respect to every cluster. The motivation behind such approach is the understanding that a point lying further apart in the space may be an anomaly or, it could simply be part of a different lower-density cluster; whereas a point lying close to a densely populated cluster may not present itself as anomalous in an absolute sense; but in a relative sense, may be a potential outlier. Precisely due to this architecture, a relative density-based clustering approach such as REDCLAN outperforms traditional methods by identifying potential outliers more accurately and robustly.

3. RELATIVE DENSITY BASED CLUSTERING AND ANOMALY DETECTION

3.1. Motivation

An important property of many real-data sets is that their intrinsic cluster structure cannot be characterized by global density parameters [13]. Very different local densities may be needed to reveal clusters in different regions of the data space. For example, in Figure 1,it is not possible to detect the clustersC_1, C_2, C_3simultaneously using one global density parameter.

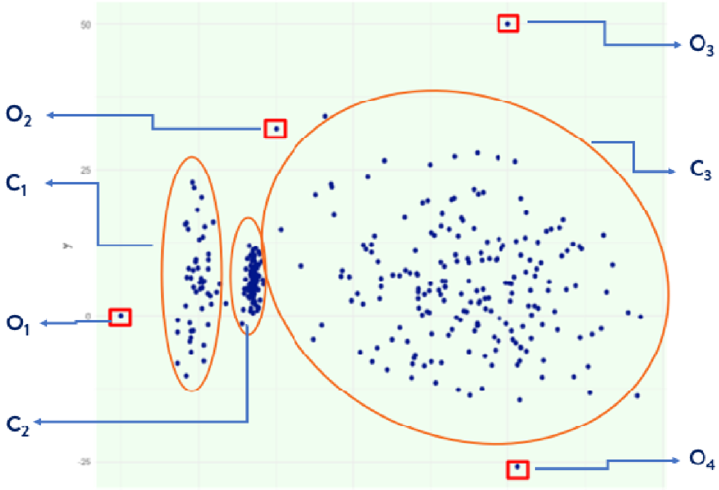

Figure 1. Data with Clusters of Varying Density

Here, we should note that this synthetic 2D dataset will be used several times for illustration purposes; this contains three Gaussian clusters of varying density $\{C_1, C_2, C_3\}$and 4 deliberately introduced outliers $\{O_1, O_2, O_3, O_4\}$

The aforementioned drawback of density-based clustering techniques (such as DBSCAN) can be understood as following: if the density of the high density cluster $C_2$is taken as the global density parameter then many a points in $\{C_1, C_3\}$will be seen as noise points – as they have (relatively) lower density, on the other hand if density of $C_3$is taken as the global density parameter then C_2 will be over-fragmented by algorithms such as DBSCAN [12], as is evident in the following Figure2.

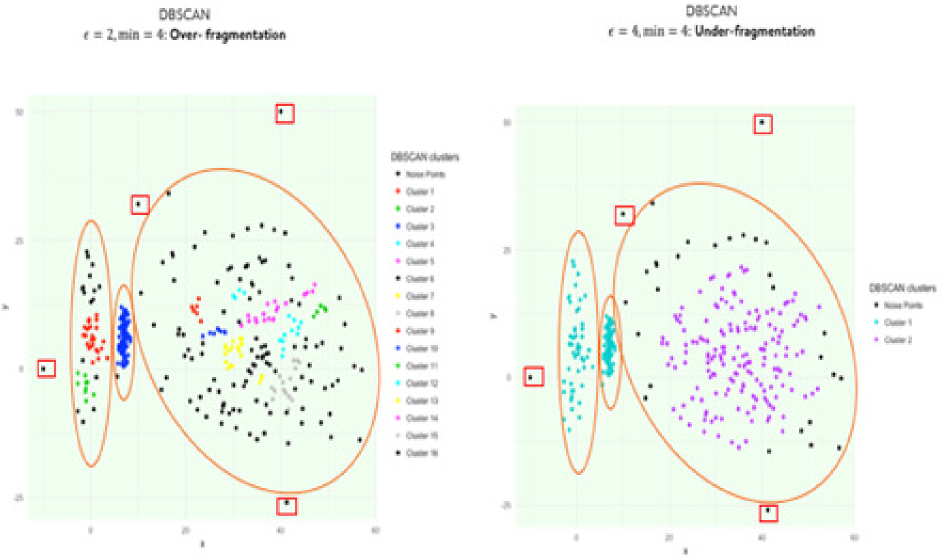

Figure 2. Data with Clusters of Varying Density: Use of DBSCAN

To overcome this problem, one needs to consider density relative to its neighbour, something called *relative density*, which will be formally defined later – this essentially means the density parameter for considering a set of points to be included in a cluster or to be left as noise points, will vary from point to point.

As we understand, after the clustering step we will have clusters and noise points; The natural next step for a data categorization algorithm such as ours would be to find outliers in the data; which we will accomplish by using a *weighted rank-based anomaly detection* technique. Since, the performance of outlier detection algorithms depends on how good the clustering algorithm captures the structure of cluster [20]– this algorithm provides significant improvement in datasets comprised of clusters of varying density.

3.2. Definitions

The following definitions will be used while describing the algorithm. First, consider the set of all d – dimensional points in the given data to be denoted by $\mathcal{D} = \{X_1, \dots, X_n\}$

For what follows, whenever we mention for a point p, it is understood$p \in \mathcal{D}$.Whenever distance of two points$\boldsymbol{x}, \boldsymbol{y} \in \mathcal{D}$is discussed, we assume the natural Euclidean distance: $d(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^T \boldsymbol{y}$

- **kNN – neighbourhood:** If $d_k(p)$is the distance between p and its k^{th} nearest neighbour, then denote the set of k nearest neighbors of p by
$$\mathcal{N}_k(p) = \{q \in \mathcal{D} \setminus \{p\} : d(p,q) \leq d_k(p)\}$$

- **Adaptive Bandwidth:** Suppose, for a point p, we have its kNN neighborhood $\mathcal{N}_k(p)$, and given fixed $\epsilon > 0$
$$\Delta_k(p) = max\,(d(p,q) : q \in \mathcal{N}_k(p))$$
$$\delta_k(p) = min\,(d(p,q) : q \in \mathcal{N}_k(p))$$
$$\bar{\delta}_k(p) = \frac{1}{k} \sum_{q \in \mathcal{N}_k(p)} d(p,q)$$

  we form an *adaptive bandwidth* around the point p as following

$$h(p) \; = \; \Delta_k(p) \; + \delta_k(p) \; + \epsilon \; - \backslash\bar{\delta}_k(p)$$

- **kNN-based Relative Density:** Following definition 2, a balloon estimator [21] might be defined as

$$\rho(x) = \frac{1}{n\big(h(x)\big)^d} \sum_{1}^{n} K\left(\frac{(x - X_i)}{h(x)}\right)$$

on top of this rather dynamic definition of density, we add another layer of local scaling and define our *relative density* as

$$\tilde{\rho}(x) = \frac{\rho(x)}{\frac{1}{k}\sum_{X_i \in \mathcal{N}_k(p)} \rho(x)}$$

- **Core Points:** A point will be denoted as *core point* if it has high relative density, i.e. for some threshold θ_1a point p will be denoted a core point iff

$$\tilde{\rho}(x) \geq \theta_1$$

authors typically determine $\theta_1$using bootstrap on the entire set of relative densities
The set of core points will be denoted by $\mathfrak{C}$

- **Directly Reachable:** A point p is said to be *directly reachable* from another point q iff

$$-\, q \in \mathfrak{C}$$
$$-\, p \in \mathcal{N}_k(q)$$

- **Reachable:** A point p is said to be *reachable* from another point q iff

$$\exists p_1, p_2, \dots\, , p_n with\, p_1 \; = \; q, p_n = p \; such\; that$$
$$p_{i+1} \; is\; directly\; reachable\; from\; p_i \; \forall i \; = 1,2, \dots, n-1$$

- **Connected:**A point p is said to be *connected* with another point q iff

$$\exists \; o \in \mathcal{D} \; such\; that\; bothp\; andq\; are\; reachable\; fromo$$

- **Rank:**The *rank* [22] of p w.r.t. q is defined as

$$rank_q(p) \; = \; |\{X_i \in \mathcal{D}: d(q, X_i) \leq \; d(p,q)\}|$$

in informal terms, this is the order rank of p w.r.t. $q$quantified by the number of points between q and p plus 1

- **Outlierness:**Outlierness of a point is a function of the weighted sum of its rank w.r.t. its neighbour. Suppose $q \in \mathcal{N}_k(p)$. Define weights by

$$w(q) = \begin{cases} \dfrac{1}{|\mathcal{C}|}, & q \in \mathcal{C} \\ \quad 1, & q\; is\; noise\; point \end{cases}$$

Essentially, if q is part of a cluster $\mathcal{C}$ define, $w(q) = \frac{1}{|\mathcal{C}|}$ if q is not part of any cluster (noise point) then $w(q) = 1$. This is to say every cluster has weight 1 which is equally divided among its components.

With this weightage scheme in hand *Outlierness* will be defined as

$$O(p) = \frac{\sum_{q \in \mathcal{N}_k(p)} w(q) rank_q(p)}{k}$$

3.3. Methodology

Next, we will discuss the algorithm in detail and why/how it works. It might be helpful to demonstrate the anatomy of the algorithm by using it on the synthetic dataset shown in Fig 1.

I. **Core Point Detection:** The very first step would be to find the set of core points $\mathfrak{C}$, this is done with the help of definition 4. Since, the core points are defined based on relative density and not absolute density – we can note (as in Fig 3), the core points will be spread across all the clusters – both dense and sparse ones, this forms the backbone of our algorithm.

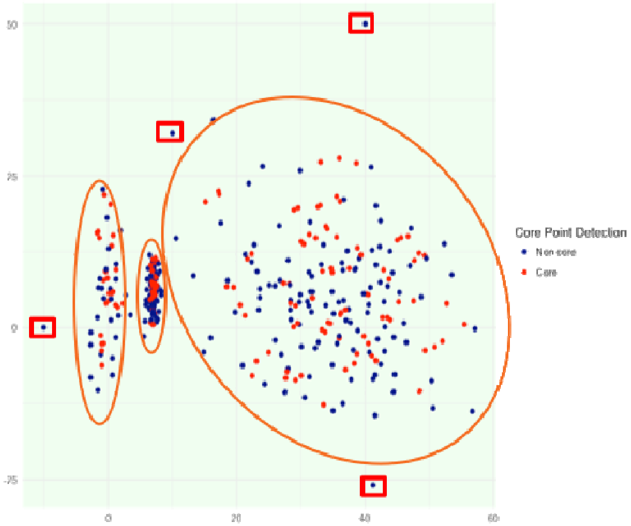

Figure 3. Core point detection

II. **Clustering:**Given the set of core points$\mathfrak{C}$, we will cluster the points into separate clusters. The clustering logic will be the following logic.

Define, $\Psi: \mathcal{D} \rightarrow \mathbb{N}$to be the clustering function which assigns a cluster number to every point in $\mathcal{D}$

Also denote,

$$n = |\mathcal{D}|$$

$\mathcal{A}(p)$: union of p and the set of points directly reachable from p

Initialize:

$$\Psi(p) \leftarrow 0 \; \forall p \in \mathcal{D}$$

$$C \leftarrow \max(\Psi)$$

```
**while**$p \in \mathcal{D}$**do**:
    **if** $\Psi(p) = 0$**then**
        **if** $p \in \mathfrak{C}$**then**
                $\Psi(\mathcal{A}(p)) \leftarrow C + 1 \; \forall p \in \mathcal{D}$
                        $C \leftarrow \mathrm{C} + 1$
        **end if**
    **end if**
**end while**
```

At the end of this step(Fig 4) points are either put into a cluster or left as noise points. $\Psi(p)$ is the cluster number a point is assigned. If $\Psi(p) = 0$, this means the point is left as a noise point.

This is a one-time breadth-first process and depends on two input parameters, k_1, k_2.
$\boldsymbol{k_1}$**:** the k used in determining adaptive bandwidth in step I (core point detection), the higher the value of $k_1$the lower number of core points will be found, and more and more core points will be concentrated towards the denser cluster.

$\boldsymbol{k_2}$: the k used in determining the reachability of the points in step II, the higher the value of $k_2$fewer clusters will be found.

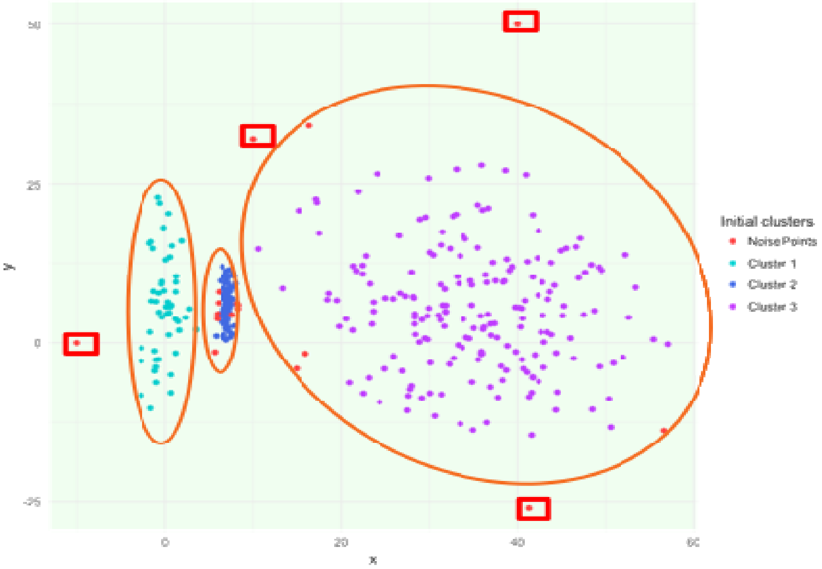

Figure 4. Clustering

III. **Anomaly Detection:** At the end of step II, we have clusters and noise points – these noise points maybe either anomalies or just boundary of given clusters, so we will call them potential outliers. Our algorithm goes the extra mile by finding outliers from the set of potential outliers using the weighted rank-based anomaly detection method.

Now, for a suitable threshold, θ_2 we do the following:

```
**while**$p \in \mathcal{D}$**do**:
        **if** $\Psi(p) = 0$**then**
                $O(p) = \frac{\sum_{q \in \mathcal{N}_k(p)} w(q) rank_q(p)}{k}$
                **if**$O(p) \geq \theta_2$**then**
                        $\Psi(p) \leftarrow -1$
                **end if**
        **end if**
**end while**
```

Note, authors have used $\boldsymbol{k = k_2}$in determining neighbourhood while calculating Outlierness.

At the end of this step (Fig 5), If $\Psi(p) = -1$, this means the point is assigned an **outlier**status.

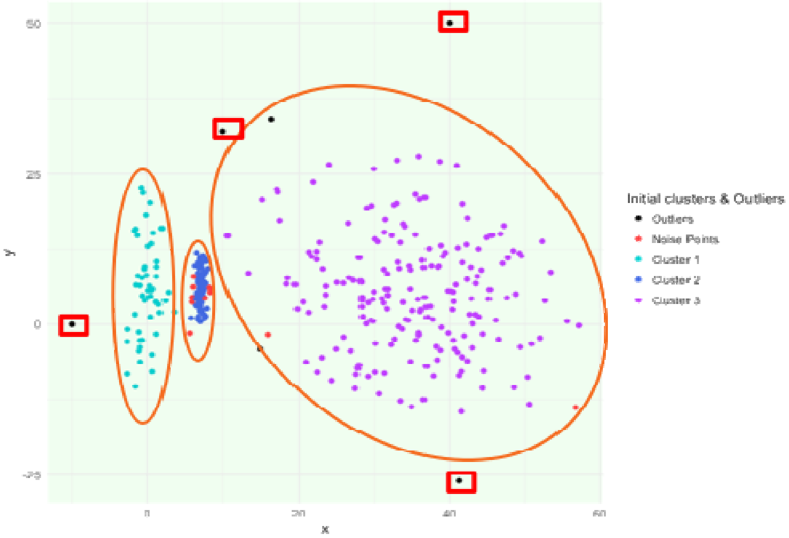

Figure 5. Outlier Detection

IV. **Cluster Proposal:**In this *optional* final step, we wrap up the process by suggesting a cluster for the set of points $\{p: \Psi\ (p) = 0\}$, i.e. points which are noise points but not anomalies. User may choose to run this step, or may ignore it and keep non-anomalous noise points as is based on application requirement

This is a rather easy task as we already have somewhat labelled scenario. Let, $\mathcal{C}_j$ denote the j^{th}cluster – We do the following:

```
**while**$p \in \mathcal{D}$**do**:

        **if** $\Psi(p) = 0$**then**
                **for** $j\ in\ 1\ to\ max(\Psi)$**do**:
                        $\delta_p(j) = \frac{1}{|\mathcal{C}_j|} \sum_{q \in \mathcal{C}_j} d(p,q)$
                **end for**
                **if**$l = argmin\ (\delta_p)$**then**
                $\Psi(p) \leftarrow l$
            end if
        end if
end while
```

As we see (Fig 6.) this clears up the remaining points by assigning them a cluster closest to them. This concludes our algorithm.

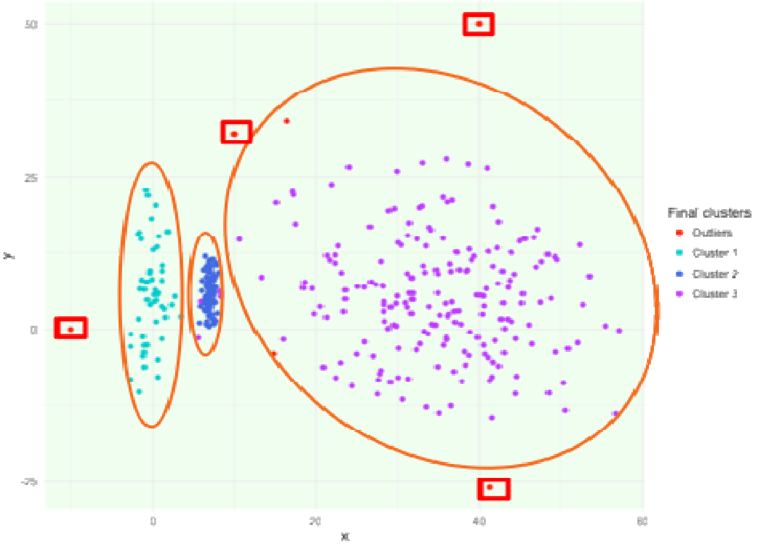

Figure 5. Final Cluster Proposal

4. EXPERIMENTS

In this section we show results of the experiments we have performed over two 2D synthetic datasets, only 2D datasets is demonstrated here partly because similar data sets have been used most extensively by other authors [23] and partly because it is easy to evaluate the quality of clustering on 2D data sets by naked eye – hence these are better suited for space constrained scenarios such as these.

4.1. Benchmarks

For comparison purposes, we will be using two other algorithms. First is DBSCAN, which is probably the most renowned and most used density-based clustering algorithm. Second is SNN based clustering proposed in [24], which has shown empirical superiority over similar methods such as k-means, DBSCAN, CURE [25] etc. We have chosen these two algorithms as the anatomy of these two matches with our algorithm – as all three revolve around the idea of identifying core points and building clusters around them. All of these do not require number of clusters to be defined by user and works better than other methods when applied on spatial data. However, as mentioned earlier: While DBSCAN can find clusters of arbitrary shapes, it cannot handle data containing clusters of differing densities, since its density-based definition of core points cannot identify the core points of varying density clusters; something that SNN seems to alleviate, proving to be superior in terms of identifying clusters of widely different shapes, sizes, and densities.

Authors would like to emphasize, since the parametrization of all three algorithms used here are very fluid and different values of parameters provide vastly different results, we have experimented with a wide array of inputs for all of the algorithms, and will only be sharing the best outcomes for individual algorithms and their corresponding input values.

4.1. Synthetic Dataset 1

Coming first is the dataset we have used for illustration purposes throughout the paper; Figures 6, 7show how DBSCAN and SNN perform on this dataset respectively

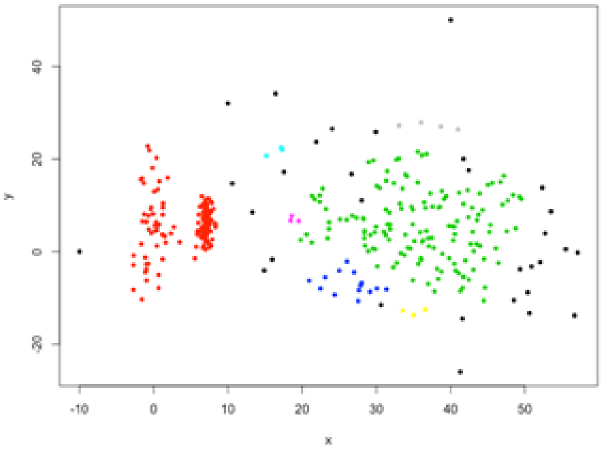

Figure 6. DBSCAN on Dataset1: $\epsilon = 3; Minpt = 3$

We can see even at its best, DBSCAN fails miserably – over-fragmenting C_3 which is the low-density cluster and mixing the other two higher-density cluster together all the while creating plenty of noise point for the user to deal with.

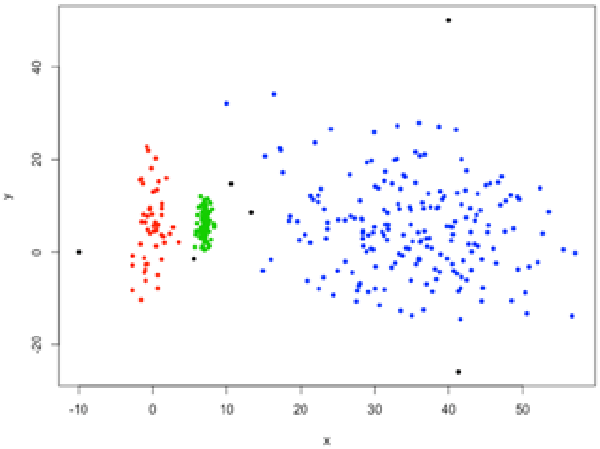

Figure 7. SNN on Dataset1: $\epsilon = 3; Minpt = 3; k = 12$

SNN on the other hand is quite adept at handling clusters with varying density identifying the clusters near perfectly, however, it fails to observe the anomalies, labelling some points as anomalies when they are actually part of clusters and failing to identify 1 out of the 4 outliers. However, one should acknowledge SNN as a huge improvement over more traditional DBSCAN.

Fig 8 shows REDCLAN almost perfectly identifies every cluster and also recognises 4 (and only 4) outliers in the dataset, only mis-classifying one point which as a boundary point. This dataset provides a great case study - on one hand without doubt our algorithm surpasses DBSCAN, it also enjoys a unique edge over algorithms such as SNN which do correct for varying density but don't have any way of differentiating between noise points created and actual outliers. This makes REDCLAN somewhat Swiss army knife for data mining tasks - which is reflected not only in quality of result but also user-friendliness and satisfaction.

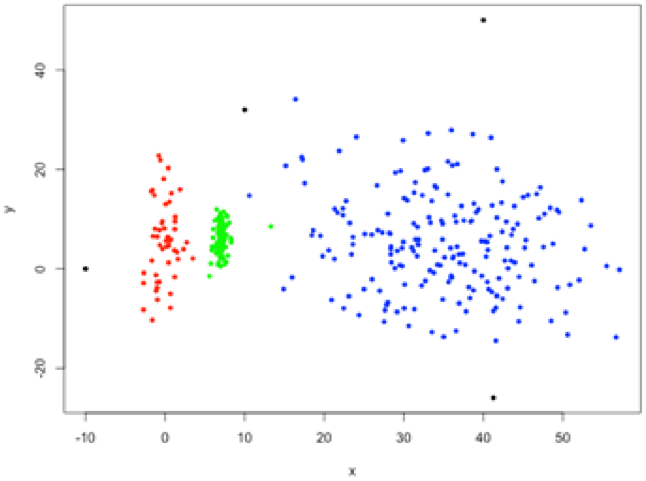

Figure 8. REDCLAN on Dataset1: $k_1 = 4; k_2 = 11$

4.3. Synthetic Dataset 2

This dataset (Fig 9) was originally part of CHAMELEON [23] study and is publicly available as part of the R package 'seriation' [26]. We can see 8 different clusters of vastly different shape, size and density floating in a pool of noise points. This appears to be a comprehensive test of competence for algorithms working on low-dimensional spatial data.

The results of the algorithms on this data can be viewed in Figures10,11,12.

Again, one can note similar results and a clear hierarchy of proficiency among the three algorithms, DBSCAN when faced with various degrees of densities gives unsatisfactory results; unnecessarily creating smaller clusters in a low-density cluster and merging two higher density cluster just as earlier. SNN,as expected, performs better than DBSCAN at least in terms of identifying lower – density clusters correctly. However, it falls short of the accuracy it gained in the previous dataset. In fact, we can see merging of higher – density cluster here too: possibly due to inability to adapt to such changes in density in the data. Moreover, the pool of noise points creates problems for SNN; it ends up creating small inconsequential clusters among these points.
The ability of REDCLAN in dealing with all these issues can be demonstrated here. It again outperforms the other two methods by pinpointing the 8 clusters and the surrounding noise points. One can notice however, few noise points are assigned a cluster number - this is due to the fact that they are so close to the cluster spatially, they almost act as boundary points.

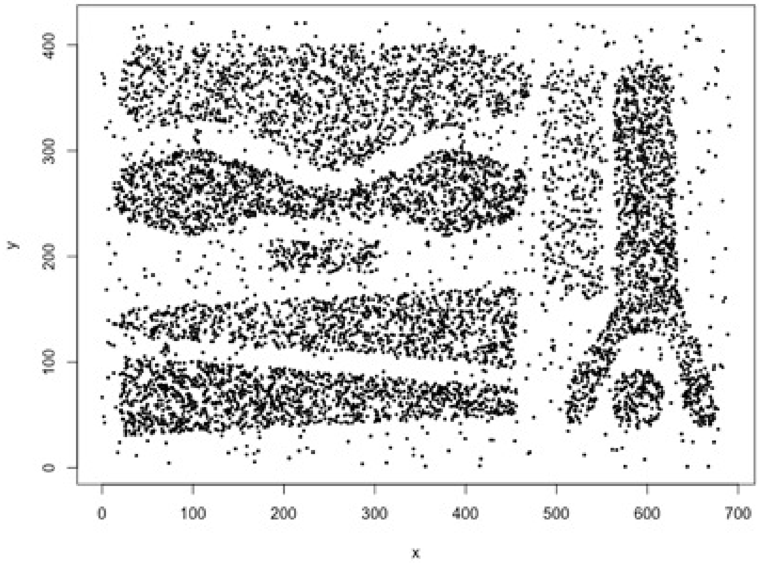

Figure 9. CHAMELEON dataset

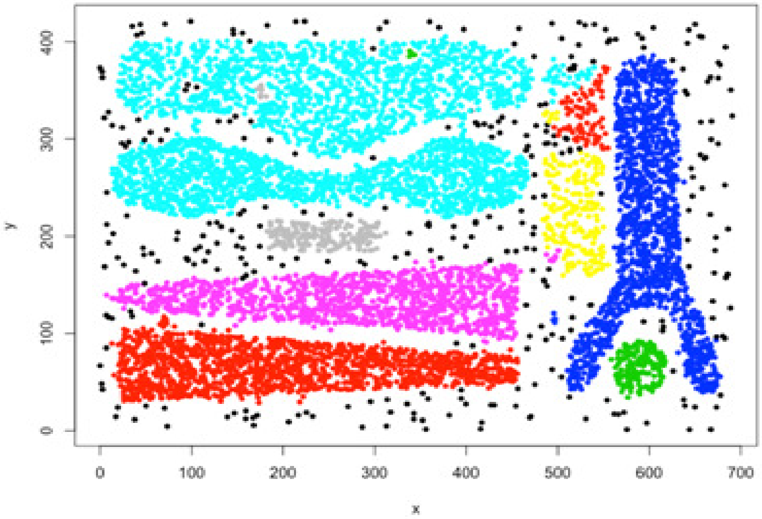

Figure 10. DBSCAN on Dataset1: $\epsilon = 8; Minpt = 4$

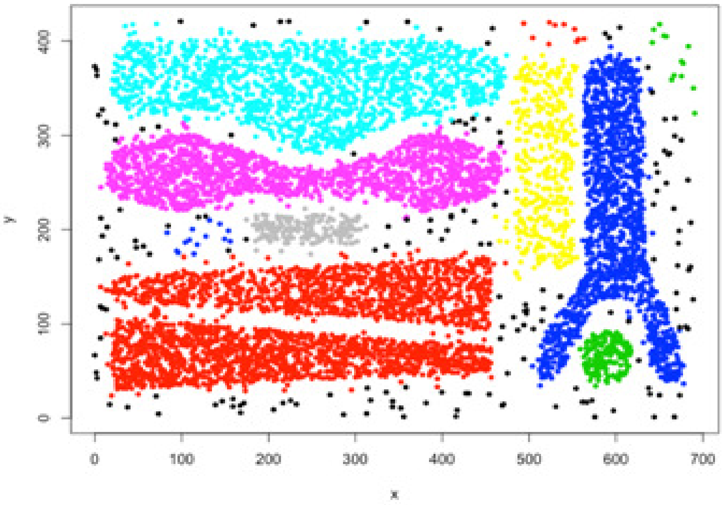

Figure 11. SNN on Dataset2: $\epsilon = 5; Minpt = 10; k = 15$

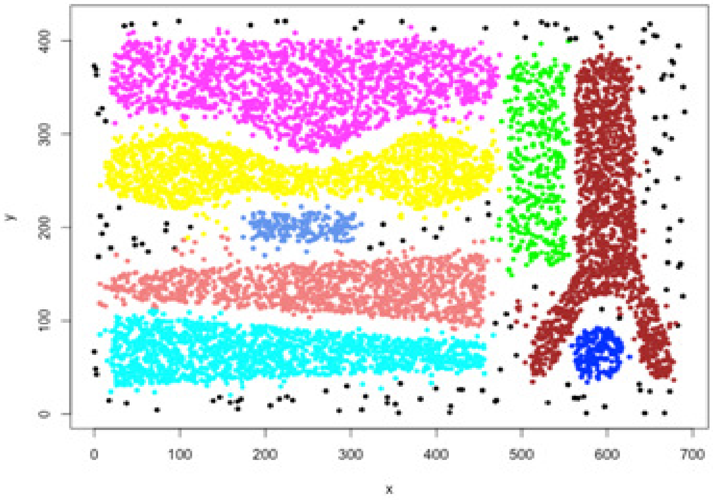

Figure 12. REDCLAN on Dataset2: $k_1 = 35; k_2 = 14$

5. CONCLUSIONS

In this paper, we present a novel technique of clustering and anomaly detection where both work in a complementary fashion. We have established the case for identification of varying density clusters which is the most practical case owing to the multifarious nature of the data. Our methodology shows notable improvements over previous density-based clustering methods like DBSCAN and SNN which are popularly used. Even though we have demonstrated the performance on synthetic datasets for the sake of comparison with previous methods, our technique particularly becomes effective while dealing with various problems in e-commerce and finance. Identifying various minute classes of substitutes or finding database anomalies from a large streaming data or identifying anomalous behaviour in the buyer-seller network are some of the prominent use-cases where our method has seen success.

REFERENCES

[1] A. K. Jain and R. C. Dubes, Algorithms for Clustering Data, Prentice-Hall, Inc., 1988.

[2] D. c. a. review, "A.K. Jain; M.N. Murty; P.J. Flynn," ACM COMPUTING SURVEYS, vol. 31, no. 3, 1999.

[3] Y. Zhao and G. Karypis, "Data clustering in life sciences," Molecular Biotechnology, vol. 31, no. 1, pp. 55-80, 2005.

[4] F. H. Bprgen and D. C. Barnett, "Applying Cluster Analysis in Counseling Psychology Research," Journal of Counseling Psychology, vol. 34, no. 4, pp. 456-468, 1987.

[5] J. Han, M. Kamber and J. Pei, Data Mining: Concepts and Techniques, San Francisco: Morgan Kaufmann Publishers Inc., 2011.

[6] P. Berkhin, "A Survey of Clustering Data Mining Techniques," in Grouping Multidimensional Data, Springer, 2006, pp. 25-71.

[7] D. Xu and Y. Tian, "A Comprehensive Survey of Clustering Algorithms," Annals of Data Science, vol. 2, no. 2, pp. 165-193, 2015.

[8] R. R. ,. L. T Zhang, "BIRCH: an efficient data clustering method for very large databasesss," in ACM Sigmod Record, 1996.

[9] S. Guha, R. Rastogi and K. Shim, "ROCK: A robust clustering algorithm for categorical attributes," Information systems, vol. 25, no. 5, pp. 345-366, 2000.

[10] MacQueen and James, "Some methods for classification and analysis of multivariate observations," in Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Oakland, 1967.

[11] H.-S. Park and C.-H. Jun, "A simple and fast algorithm for K-medoids clustering," Expert systems with applications, vol. 36, no. 2, pp. 3336--3341, 2009.

[12] M. E. Xu, H.-P. Kriegel, J. Sander and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in KDD'96 Proceedings of the Second International Conference on Knowledge Discovery and Data Mining Pages, Portland, 1996.

[13] M. A. Sander, M. M. Breunig, H.-P. Kriegel and J. Sander, "OPTICS: Ordering Points To Identify the Clustering Structure," in International Conference on Management of Data, 1999.

[14] V. Chandola, A. Banerjee and V. Kumar, "Anomaly Detection: A Survey," ACM Computing Surveys, vol. 41, no. 3, 2009.

[15] J. Laurikkala, "Informal Identification of Outliers in Medical Data," in Proceedings of the Fifth International Workshop on Intelligent Data Analysis in Medicine and Pharmacology, Berlin, 2002.

[16] F. E. Grubbs, "Procedures for Detecting Outlying Observations in Samples," Technometrics, 1969.

[17] P. J. Rousseeuw and A. M. Leroy, Robust Regression and Outlier Detection, New York: John Wiley & Sons, Inc., 1996.

[18] H.-P. K. R. T. N. J. S. Markus M. Breunig, "LOF: Identifying Density-Based Local Outliers," in ACM SIGMOD, Dallas, 2000.

[19] V. J. Hodge and J. Austin, "A Survey of Outlier Detection Methodologies.," Artificial Intelligence Review, vol. 22, no. 2, p. 85–126, 2004.

[20] I. Syarif, A. Prugel-Bennett and G. Wills, “Unsupervised clustering approach for network anomaly detection,” Communications in Computer and Information Science, vol. 293, 2012.

[21] G. R.Terrell and D. W. Scott, “Variable Kernel Density Estimation,” The Annals of Statistics, pp. 1236-1265, 1992.

[22] H. Huang, “Rank Based Anomaly Detection Algorithms,” Electrical Engineering and Computer Science - Dissertations.331, 2013.

[23] G. K. Kumar, E.-H. (. Han and V. Kumar, “CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling,” Computer, vol. 32, no. 8, pp. 68-75, 1999.

[24] L. E. a. M. S. a. V. Kumar, “Finding Clusters of Different Sizes, Shapes, and Densities in Noisy, High Dimensional Data,” in SDM, 2003.

[25] S. Guha, R. Rastogi and K. Shim, “Cure: an efficient clustering algorithm for large databases,” Information Systems, vol. 26, no. 1, pp. 35-58, 2001.

[26] M. Hahsler, K. Hornik and C. Buchta, “Getting things in order: An introduction to the R package seriation,” Journal of Statistical Software, vol. 25, no. 3, pp. 1-34, 2008.

[27] A. K. a. D. R. C. Jain, Algorithms for clustering data, Prentice-Hall, Inc., 1988.

[28] X. Xu, M. Ester, H.-P. Kriegel and J. Sander, “A distribution-based clustering algorithm for mining in large spatial databases,” in Data Engineering, 1998. Proceedings., 14th International Conference on, 1998.

[29] J. A. Hartigan and M. A. Wong, “Algorithm AS 136: A k-means clustering algorithm,” Journal of the Royal Statistical Society. Series C (Applied Statistics), vol. 28, no. 1, pp. 100-108, 1979.