

# KDDClus: A Simple Method for Multi-Density Clustering

Sushmita Mitra<sup>1</sup> and Jay Nandy<sup>1</sup>

<sup>1</sup>Machine Intelligence Unit, Indian Statistical Institute, Kolkata 700 108, INDIA,  
[sushmita@isical.ac.in](mailto:sushmita@isical.ac.in)

**Abstract.** Automated clustering of multi-density spatial data is developed. The algorithm KDDClus serves as an enhancement to the well-known DBSCAN. Averaging the distances of a pattern to all  $k$  of its nearest neighbours allows a smoothing out of noise while automatically detecting the “knees” from the  $k$ -distance plot. The use of the KD-tree data structure enables efficient computation of the  $k$ -nearest neighbours ( $k$ -NN) of a pattern point, particularly for large data. Experimental results on synthetic data, involving nested multiple densities of different shapes, demonstrates the superiority of KDDClus.

**Keywords:** Density-based clustering, DBSCAN.

## 1 Introduction

Often we come across spatial data consisting of a mixture of pattern distributions involving different densities, which may or may not be nested within each other, in the presence of background noise. Clusters of different densities can, therefore, be modeled as belonging to point processes having different intensities. Clustering of such data is a challenging problem in data mining [6]. It becomes imperative to detect the number of point processes (or cluster type of a certain density) while also assigning the patterns to these different clusters. One needs to estimate a number of thresholds in order to discriminate between these different density distributions. Automatic estimation of such parameters is a difficult task. The complexity of searching the neighbourhood is also large in high-dimensions.

The density-based approach addresses this issue, while detecting clusters of differing densities having arbitrary shape and size. It is non-parametric, and requires no prior information regarding the number of clusters or their underlying density. The algorithms detect the difference in densities among regions of contiguous patterns in a spatial database, and accordingly assign them to different clusters. Noise and outliers are treated as low-density regions, and are removed in terms of certain density criteria. The earliest research in this direction was reported in Refs. [4, 9]. Some of the interesting studies of efficient density-based clustering, in the context of databases and large datasets, are DBSCAN [3], OPTICS [2], DENCLUE [5], CLIQUE [1] and WaveCluster [8].

Typically these algorithms require user-specification of certain parameters, related to density-level thresholds, to be provided as input. Often this becomes all the more difficult when clusters in different regions of the feature space have considerably different densities or clusters with different density levels are nested. In such cases the partitioning might not be proper with one single density threshold.

In this article we describe a new and simple algorithm KDDClus which clusters multiple pattern distributions of different densities in the presence of noise. It is able to distinguish between different density regions, which may or may not be nested and are generally of non-convex shape. The algorithm automatically estimates a number of thresholds to optimally identify the different density regions, without any prior knowledge about the data. While conventional density-based clustering algorithms like DBSCAN typically resort to visual determination of a single threshold to distinguish between two density regions, algorithm KDDClus may be considered as an enhancement to it. The space-partitioning KD-tree data structure [7] is utilized to efficiently determine the  $k$ -nearest neighbours ( $k$ -NN) of a pattern for large data. The sorted average  $k$ -NN distances for the patterns is clustered (i) for the purpose of smoothing out the noise and (ii) automatically determining the optimal number of density regions while minimizing a validity index. The algorithm is computationally inexpensive. The experimental results on a synthetic dataset, consisting of clusters of different densities, demonstrates the effectiveness of the algorithm.

## 2 KDDClus: An Algorithm Enhancing DBSCAN

Clustering patterns involving different densities and noise, coexisting in the same spatial dataset, requires the determination of a number of thresholds. Automatic estimation of such parameters, particularly in varying densities of multiple point processes, is a difficult task.

Algorithm DBSCAN requires proper estimation of two global parameters  $\epsilon$  and  $MinPts$ . This is highly data-dependent, and can be overestimated or underestimated by the visual and/or interactive procedure used. It may automatically lead to misplacement of patterns and even misidentification of clusters. Moreover, the algorithm does not consider the handling of a simultaneous presence of different densities, originating from different point processes in the data. Note that no single set of  $\epsilon$  and  $MinPts$  can properly cluster such a dataset. The complexity of searching the neighborhood is large in high-dimensions, thereby leading to the difficulty in determining a proper distance estimate.

We present here a new and simple way to automatically identify the number of point processes (or clusters of different densities) including noise. The algorithm utilizes the  $KD$ -tree data structure for efficient processing in high dimensions. It can simultaneously estimate the different density parameters without any prior knowledge about the data. It is also not expensive.

We compute the average of the distances of a pattern to all  $k$  of its nearest neighbors. This is unlike DBSCAN, where only the  $k$ th nearest neighbor

is considered during the distance computation. The use of the  $KD$ -tree data structure [7] enables efficient computation of  $k$ -nearest neighbors ( $k$ -NN) of a point, particularly for large data. The averaging allows a smoothing of the curve towards noise removal, for subsequent easier automated detection of density-thresholds. We plot these averaged  $k$ -distances in an ascending order, to help identify noise with relative ease. Note that patterns corresponding to noise are expected to have larger  $k$ -distance values. The aim is to determine the “knees” for estimating the set of  $\epsilon$  parameters.

A knee corresponds to a threshold where a sharp change of gradient occurs along the  $k$ -distance curve. This represents a change in density distribution amongst patterns. Any value less than this density-threshold  $\epsilon$  estimate can efficiently cluster patterns whose average  $k$ -NN distances is lower than that, implying patterns belonging to a certain density. Analogously all knees in the smoothed graph can collectively estimate a set of  $\epsilon$ 's for identifying all the clusters having different density distributions. The knee regions are detected in KDDClus by clustering the sorted  $k$ -NN distances. We determine the optimal number  $c_0$  of such segments, by using  $c$ -means while optimizing a clustering validity index.

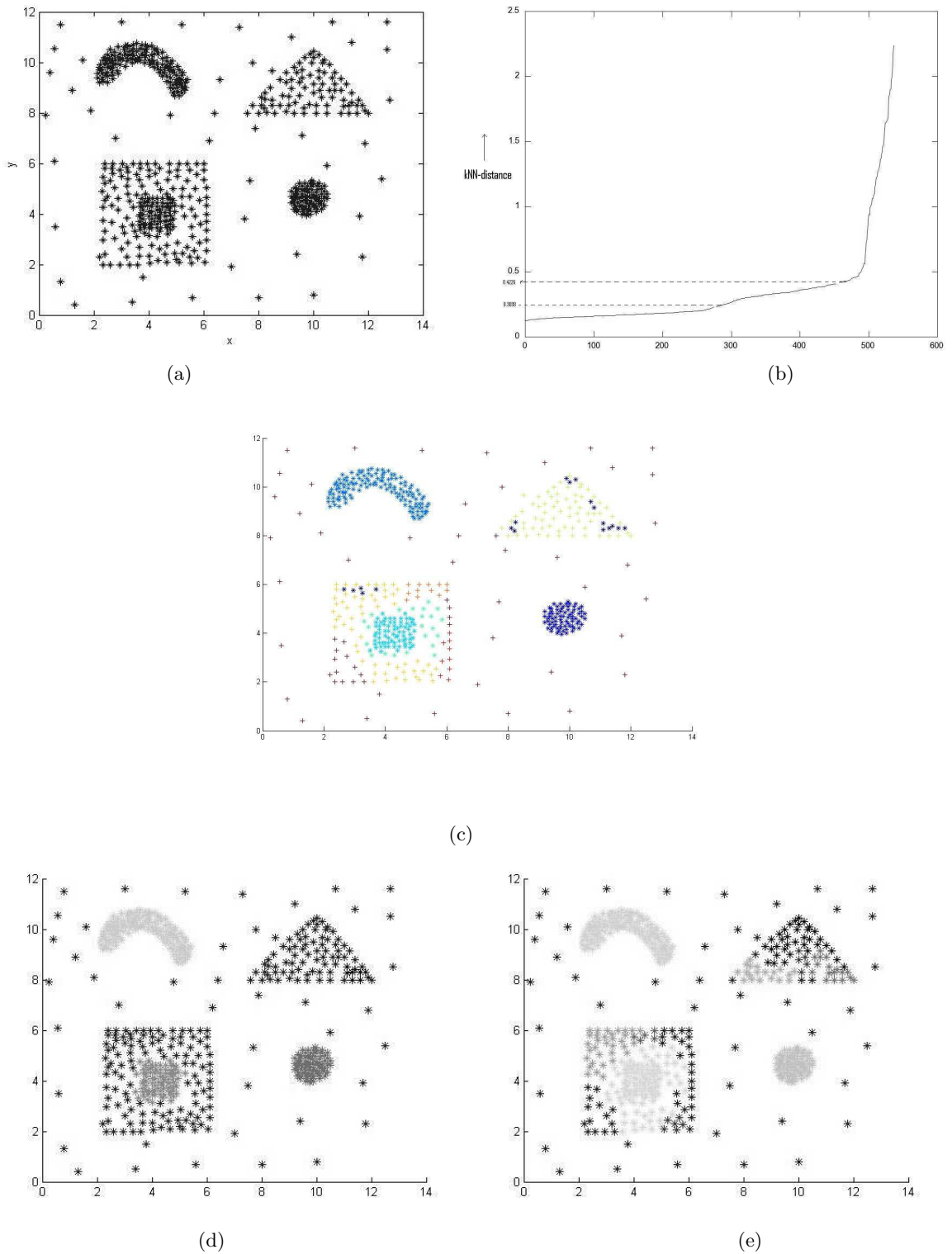
Starting from the lowest value in the sorted  $k$ -NN distance graph, we sequentially execute DBSCAN for each of the  $c_0$  estimated  $\epsilon$ 's considered in ascending order. The first estimate obviously corresponds to the most dense cluster. Tagging the patterns in the already detected clusters as “visited”, we proceed towards larger values of  $k$ -distance while allowing DBSCAN to work on the still-unvisited patterns only. In this manner we are able to effectively determine all clusters in a multi-density framework, in a decreasing order of density, with noise being modelled as the sparsest region.

### 3 Experimental Results

We have implemented the proposed KDDClus algorithm on a synthetic pattern set. There exist five clusters with three different densities for dataset *Decode* in Fig. 1(a). The semi-circular region on the top-left, inner quadrilateral, and circular region on bottom-right of the figure constitute the most dense clusters. The outer quadrilateral and triangular region form the medium-density clusters. The background is least dense and consists of noise.

We find from part (d) of the figure that DBSCAN, with the lower threshold of  $\epsilon = 0.3038$ , could correctly identify only (i) the smaller quadrilateral inscribed within the larger one, (ii) the circle, and (iii) the semi-circular region. These are indicated by different shades of gray in the figure. Using DBSCAN with the higher  $\epsilon$ -value of 0.4226 resulted in the output map in part (e) of the figure. In this case it is noticed that the smaller dense quadrilateral along with some surrounding points from the outer medium-density quadrilateral get merged into one cluster.

This adverse effect is eliminated with algorithm KDDClus, as observed from part (c) of the figure. After correctly detecting the smaller quadrilateral, the circle and the semi-circular region with  $\epsilon = 0.3038$ , the algorithm marks the



**Fig. 1.** Pattern *Decode*. (a) Original pattern, with (b) sorted average  $k$ -distance plot and (c) KDDclus clustering; DBSCAN clustering with (e)  $\epsilon = 0.3038$ , (f)  $\epsilon = 0.4226$ .

constituent patterns from the dataset as visited. In the next step DBSCAN needs to work only on the unvisited patterns with  $\epsilon = 0.4226$ . Now it is able to correctly distinguish the larger quadrilateral and the triangle as the second lower-density level, within the background noise.

## 4 Conclusions

The algorithm KDDClus is an enhancement to DBSCAN, in terms of automatically estimating the various density-based parameters for optimal clustering. Unlike DBSCAN, where only the  $k$ th nearest neighbour is considered during the distance computation, here we calculate the average of the distances of a pattern to all  $k$  of its nearest neighbours. Such averaging allows a smoothing of the curve for subsequent easier automated detection of the “knees” amongst the background noise. The use of the KD-tree data structure enables efficient computation of the  $k$ -nearest neighbours ( $k$ -NN) of a pattern point, particularly for large data. Comparative study has been made on three sets of synthetic data to establish the superiority of the proposed algorithm.

## References

1. Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic sub-space clustering of high dimensional data for data mining applications. In: Proceedings of 1998 ACM-SIGMOD International Conference on Management of Data (SIGMOD'98). pp. 94–105. Seattle, USA (June 1998)
2. Ankerst, M., Breunig, M., Kriegel, H.P., Sander, J.: OPTICS: Ordering points to identify the clustering structure. In: Proceedings of 1999 ACM-SIGMOD International Conference on Management of Data (SIGMOD'99). pp. 49–60. Philadelphia, USA (June 1999)
3. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases. In: Proceedings of 1996 International Conference on Knowledge Discovery and Data Mining (KDD'96). pp. 226–231. Portland, USA (August 1996)
4. Hartigan, J.A.: Clustering Algorithms. John Wiley & Sons (1975)
5. Hinneburg, A., Keim, D.A.: An efficient approach to clustering in large multimedia databases with noise. In: Proceedings of 1998 International Conference on Knowledge Discovery and Data Mining (KDD'98). pp. 58–65. New York, USA (August 1998)
6. Mitra, S., Acharya, T.: Data Mining: Multimedia, Soft Computing, and Bioinformatics. John Wiley, New York (2003)
7. Moore, A.: A tutorial on  $KD$ -trees. Computer Laboratory Technical Report # 209, University of Cambridge, <http://www.cs.cmu.edu/~awm/papers.html> (1991)
8. Sheikholeslami, G., Chatterjee, S., Zhang, A.: WaveCluster: A multi-resolution clustering approach for very large spatial databases. In: Proceedings of 1998 International Conference on Very Large Data Bases (VLDB'98). pp. 428–439. New York, USA (August 1998)
9. Wishart, D.: Mode analysis: A generalization of nearest neighbor which reduces chaining effects. Numerical Taxonomy (1969)