Locally Scaled Density Based Clustering

Ergun Biçici and Deniz Yuret

Koç University
Rumelifeneri Yolu 34450
Sariyer Istanbul, Turkey
{ebicici,dyuret}@ku.edu.tr

Abstract. Density based clustering methods allow the identification of arbitrary, not necessarily convex regions of data points that are densely populated. The number of clusters does not need to be specified beforehand; a cluster is defined to be a connected region that exceeds a given density threshold. This paper introduces the notion of local scaling in density based clustering, which determines the density threshold based on the local statistics of the data. The local maxima of density are discovered using a k-nearest-neighbor density estimation and used as centers of potential clusters. Each cluster is grown until the density falls below a pre-specified ratio of the center point's density. The resulting clustering technique is able to identify clusters of arbitrary shape on noisy backgrounds that contain significant density gradients. The focus of this paper is to automate the process of clustering by making use of the local density information for arbitrarily sized, shaped, located, and numbered clusters. The performance of the new algorithm is promising as it is demonstrated on a number of synthetic datasets and images for a wide range of its parameters.

1 Introduction

Clustering is the process of allocating points in a given dataset into disjoint and meaningful clusters. Density based clustering methods allow the identification of arbitrary, not necessarily convex regions of data points that are densely populated. Density based clustering does not need the number of clusters beforehand but relies on a density-based notion of clusters such that for each point of a cluster the neighborhood of a given radius (ε) has to contain at least a minimum number of points ($\wp$). However, finding the correct parameters for standard density based clustering [1] is more of an art than science.

This paper introduces the locally scaled density based clustering (LSDBC) algorithm, which clusters points by connecting dense regions of space until the density falls below a threshold determined by the center of the cluster. LSDBC takes two input parameters: k, the order of nearest neighbor to consider for each data point for density calculation and α, which determines the boundary of the current cluster expansion based on its density. The algorithm is robust to background noise and density gradients for a wide range of its parameters.

Density based clustering in its original form, DBSCAN [1], is sensitive to minor changes in its parameters known as the neighborhood of a given radius (ε) and the minimum number of points that need to be contained within the neighborhood ($\wp$). We discuss density based clustering and identify some of its drawbacks in Section 2. Although using different parameters for the radius of the neighborhood and the number of points contained in it appear to give some flexibility, these two parameters are actually dependent on each other. Instead, the LSDBC technique employs the idea of local scaling. We order points according to their distance to their kth neighbor. This gives an approximate measure of how dense the region around each point is. Then, starting with higher density points, we cluster densely populated regions together. The resulting clustering technique does not require fine tuning of parameters and is more robust. OPTICS [2] also bases its clustering decisions on the local density by using *kNN* type density estimation (differences are explored in section 6).

The local scaling technique, previously employed successfully by spectral clustering [3], makes use of the local statistics of points to separate the clusters within the dataset. The idea is to scale each point in the dataset with a factor proportional to its distance to its kth neighbor. Section 3 discusses local scaling and how it can be used for clustering purposes. We show that when local scaling is used in density based clustering, it creates more robust clusters and allows the automatic creation of clusters without any need for parameters other than k, the order of nearest neighbor to consider, and α, which decides when the drop in the density is necessary for the cluster change.

Density based clustering is important for knowledge discovery in databases. Its practical application aresas include biomedical image segmentation [4], molecular biology and geospatial data clustering [5], and earth science tasks [1].

The following lists the contributions of this paper. We introduce locally scaled density based clustering (section 4), which correctly ignores background clutter and identifies clusters within background noise. LSDBC is also robust to changes in the parameters and produces stable clusters for a wide range of them. LSDBC makes the underlying structure of high-dimensional data accessible. The problems we deal with include: (1) finding appropriate parameter values, (2) handling data with different local statistics, (3) clustering in the presence of background clutter, and (4) reducing the number of parameters used. Our results show better performance than prominent clustering techniques such as DBSCAN, k-means, and spectral clustering with local scaling on synthetic datasets (section 5). Our results on image segmentation tasks also show that LSDBC is able to handle image data and segment it into meaningful regions. Related work and density estimation are discussed in section 6 and the last section concludes.

2 Density Based Clustering

Density based clustering differentiates regions which have higher density than its neighborhood and does not need the number of clusters as an input parameter. Regarding a termination condition, two parameters indicate when the expansion of clusters should be terminated: given the radius of the volume of data points to look for, ε, a minimum number of points for the density calculations, $\wp$, has to be exceeded.

Let $d(p, q)$ give the distance between two points p and q; we give the basic terminology of density based clustering below. ε neighborhood of a point p is denoted by $N_\varepsilon(p)$ and is defined by $N_\varepsilon(p) = \{q \in \mathit{Points} \mid d(p, q) \leq \varepsilon\}$, where *Points* is the set of points in our dataset. A *core point* is defined as a point above the density threshold wrt. ε and $\wp$, i.e. $|N_\varepsilon(p)| \geq \wp$. A border point is defined as a point below the threshold but that belongs to the ε neighborhood of a core point.

Definition 1 (Directly density-reachable). *A point p is directly density reachable from a point q wrt. ε and $\wp$, if $p \in N_\varepsilon(q)$ and $|N_\varepsilon(q)| \geq \wp$ (core point condition).*

Definition 2 (Density-reachable).
A point p is density reachable from a point q wrt. ε and $\wp$, if there is a chain of points $p_1, p_2, ..., p_n$, $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density reachable from p_i.

Definition 3 (Density-connected).
A point p is density connected to a point q wrt. ε and $\wp$, if there is a point r such that both p and q are density reachable from r wrt. ε and $\wp$.

A *cluster* C wrt. ε and $\wp$ is a non-empty set of points such that $\forall p, q \in C$, p is density connected to q wrt. ε and $\wp$.

Selecting appropriate parameters, ε and $\wp$, is difficult in DBSCAN and even in the best setting, the results may not be good. Figure 1 gives representative results using our synthetic datasets. Note that minor changes in the parameters ε and $\wp$ creates spurious clustering results.

Ester *et al.* [1] suggest that the user will look at the sorted 4-dist graph (plot of points' distance to their 4th nearest neighbor in descending order) and select a threshold point, which will divide the points into two sets: noise and clusters. The selected threshold, *4NNDist* value, can be used for determining the parameters as in: $\varepsilon =$ *4NNDist* and $\wp = 4$. However, for some datasets, the threshold point may not be easy to pick, it may not be unique if there is variance in the density, $k = 4$ may not be the ideal setting, and this approach assumes user intervention.

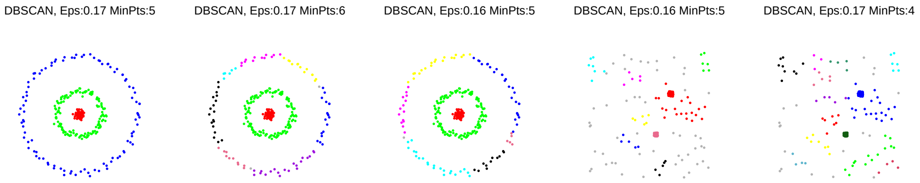


Fig. 1. Density based clustering is sensitive to minor changes in ε and $\wp$.

3 Local Scaling

Zelnik-Manor and Perona [3] successfully applied local scaling to spectral clustering. Local scaling is a technique which makes use of the local statistics of the data when identifying clusters. This is done by scaling the distances around each point in the dataset with a factor proportional to its distance to its kth nearest neighbor. As a result, local scaling finds the scale factors for clusters with different densities and creates an affinity matrix in which the affinities are high within clusters and low across clusters.

Given two points x_i and x_j from a dataset, X, let A_{x_i,x_j} denote the affinity between the two points, showing how similar two objects are. Based on [6], $\forall x_i, x_j \in X$, let the following properties hold:

$$A_{x_i,x_j} \in [0,1],\ A_{x_i,x_i} = 1,\ A_{x_i,x_j} = A_{x_j,x_i}. \tag{1}$$

We could define A_{x_i,x_j} as:

$$A_{x_i,x_j} = exp(-\frac{d^2(x_i,x_j)}{\sigma^2}), \tag{2}$$

where $d(x_i,x_j)$ is any distance function (such as the Euclidean ($||x_i - x_j||^2$) or the cosine between feature vectors) and σ is a threshold distance below which two points are thought to be similar and above which two points are considered dissimilar. A single scaling parameter, σ, may not work for the whole dataset when clusters with different densities are present. Instead, a local scaling parameter σ_i can be calculated for each data point x_i such that the affinity between a pair of points, x_i and x_j, is given by:

$$\hat{A}_{ij} = exp(-\frac{d^2(x_i,x_j)}{\sigma_i \sigma_j}), \tag{3}$$

where $d(x_i,x_j)$ corresponds to the distance from x_i to x_j. When selecting the local scaling parameter σ_i, local statistics of the neighborhood of point x_i is considered. The choice in [3] is:

$$\sigma_i = d(x_i, x_i^k), \tag{4}$$

where x_i^k is the kth closest neighbor of point x_i and k is chosen to be 7. Thus, $\sigma_i =$ *7NNDist*(x_i) in spectral clustering with local scaling.

4 Locally Scaled Density Based Clustering

Locally scaled density based clustering algorithm clusters points by connecting dense regions of space until the density falls below a threshold determined by the center of the cluster. LSDBC takes two input parameters, k, the order of nearest neighbor to consider for each point in the dataset for density calculation and α, which determines the boundary of the current cluster expansion based on its density.

The LSDBC algorithm first calculates the ε values for each point based on their *kNN* distances. ε allows us to order points based on their density. Smaller ε values correspond to denser regions in the dataset. The set of points are then sorted in ascending order of their ε. Algorithm 1 presents the main method of LSDBC. The function *kNNDistVal* takes a point and a number k and returns the distance of the point to its kth nearest neighbor, ε, as well as the set of its k nearest neighbors. *localMax* function ensures that the selected point is the most dense point locally in its neighborhood.

The *ExpandCluster* procedure, given in Algorithm 2, expands the cluster of a given point, p, by exploring neighboring points and placing them into the same cluster as p when their density is above $\frac{\textit{density}(p)}{2^\alpha}$. The initial point p is called the center point of the cluster. The α parameter prevents the expansion of a given cluster into regions of points with a density smaller than a factor of $2^{-\alpha}$ relative to the center. The density of a given point p is calculated as:

$$\textit{density}(p) = \frac{k}{\varepsilon^n}, \tag{5}$$

```
Input: $D$: Distance matrix, $k$: input to kNN-dist function, $n$: number of dimensions, $\alpha$.
Output: Allocation of points to clusters.

for $p \in$ Points do
    $p$.class $=$ UNCLASSIFIED;
    $[p.Eps, p.neighbors] =$ kNNDistVal$(D, p, k)$;
end
Points.sort();                                        /* Sort on Eps */
ClusterID $= 1$;
for $p \in$ Points do
    if $p$.class $==$ UNCLASSIFIED and localMax$(p)$ then
        ExpandCluster$(p, ClusterID, n, \alpha)$;
        ClusterID $=$ ClusterID $+ 1$;
    end
end
```

Algorithm 1: LSDBC: Locally Scaled Density Based Clustering Algorithm.

where n corresponds to the dimensionality of the dataset. A point p' is defined as a *core point* if its density exceeds the density of the center point for the cluster multiplied by $2^{-\alpha}$:

$$\frac{k}{2^{\alpha}\varepsilon_p^n} \leq \frac{k}{\varepsilon_{p'}^n}. \tag{6}$$

Therefore,

$$\varepsilon_{p'} \leq 2^{\alpha/n}\varepsilon_p. \tag{7}$$

Equation 7 provides us a cutoff point when expanding cluster regions.

In the final clustering scheme of LSDBC, we need only two parameters: the k value, which is a parameter corresponding to up to which closest neighbor we should look for when clustering points, and α which takes role in identifying a cutoff density for the cluster expansion. The focus of this paper is to automate the process of clustering by making use of the local density information for arbitrarily sized, shaped, located, and numbered clusters. LSDBC enjoys good clustering results for a wide range of values for k and α. An obvious advantage of LSDBC is that it is not sensitive to background density variations and therefore, it can be used with a wide range of clustering problems.

Computational Complexity. *kNNDistVal* operates in $O(n)$ time. As it was suggested in [1], if we use a tree-based spatial index, k nearest neighbors can be retrieved in $O(log\ n)$ time. Therefore, the run-time complexity of LSDBC algorithm is $O(n\ log\ n)$, which is the same as DBSCAN's run-time complexity.

5 Experiments

We compared our algorithm, LSDBC with (1) the original density based clustering algorithm, DBSCAN, (2) spectral clustering with local scaling, and (3) k-means clustering. The results show that LSDBC's performance is superior to these three clustering techniques on the problems we analyzed. Spectral clustering with local scaling achieves comparable performance on some of the synthetic datasets. LSDBC produces robust

```
Input: point, ClusterID, $n$: number of dimensions, $\alpha$.
point.class $=$ ClusterID;
Seeds $=$ point.neighbors;
for currentP $\in$ Seeds do
    if currentP.class $==$ UNCLASSIFIED then
        currentP.class $=$ ClusterID;
    else
        Seeds.delete(currentP);
    end
end
while Seeds.length $> 0$ do
    currentP $=$ Seeds.first();
    if currentP.Eps $\leq 2^{\alpha/n} \times$ point.Eps then
        Neighbors $=$ currentP.neighbors;
        for neighborP $\in$ Neighbors do
            if neighborP.class $==$ UNCLASSIFIED then
                Seeds.append(neighborP);
                neighborP.class $=$ ClusterID;
            end
        end
    end
    Seeds.delete(currentP);
end
```

Algorithm 2: *ExpandCluster*: Expands the cluster of a given point.

clusters for a broad range of values for k and α. The robustness of LSDBC for different values of k can be seen in Figure 2.

In this set of experiments, we compare the results of LSDBC with the clustering techniques of DBSCAN, k-means, and spectral clustering with local scaling. For each dataset, k-means and spectral clustering methods accept the number of clusters as input where $k = 20$ is the number of ideal clusters for the given dataset. Therefore, for the clustering methods that we compare, including DBSCAN, we chose the best possible setting for the clustering. In Figure 3, we compare their performance on a more complex dataset. In all of these examples, the performance of LSDBC is superior to others in terms of its ability to respect the boundaries of closely located and similarly populated regions. The difference between LSDBC's results is that when $k = 6$ or $k = 7$, the background clutter is divided into 2 clusters whereas when $k = 8$, they form a single cluster. When we look at the results of k-means and spectral clustering with local scaling, even under the best settings, we can see that they divide densely populated regions into separate clusters and merge regions with different densities together whereas DBSCAN classifies regions with lower density below a threshold as noise.

Apart from generating unnatural clusters in some datasets, another drawback of spectral clustering and k-means is their requirement of the number of clusters as input, whereas LSDBC can discover clusters of arbitrary size, shape, location, and number without any knowledge of the number of clusters in the data.

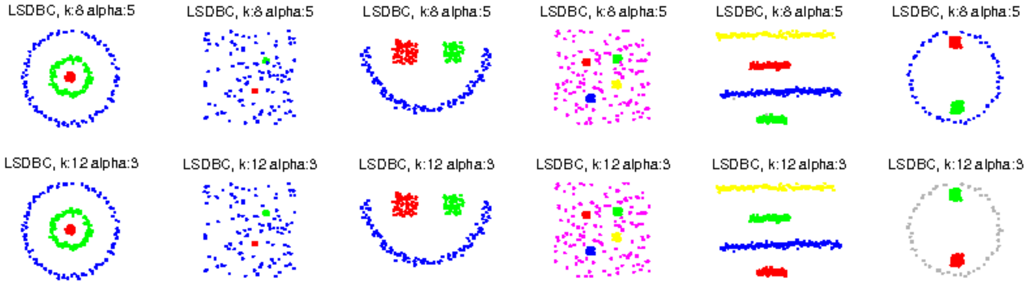


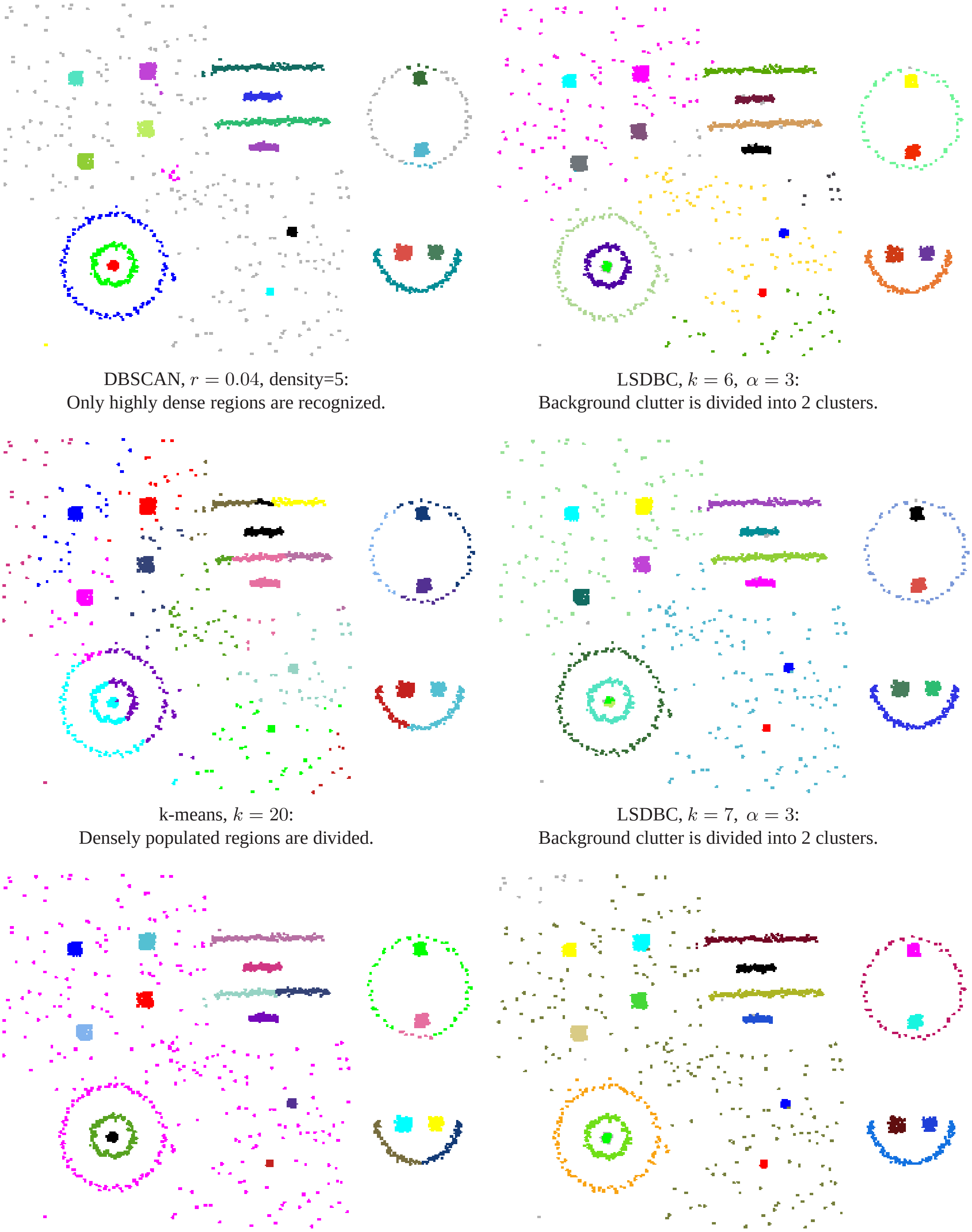
Fig. 2. Robustness of LSDBC for different values of k and α.

In our next set of experiments, we deal with the task of image segmentation. The results can be seen in Figures 4, 5, 6, and 7. LSDBC is able to decipher transparency in the images. For instance, the background seen through the handle hole in the still life image (Figure 4) and the background itself is clustered into the same cluster. The resulting clusterings can be used to summarize images as well as compress them. As can be seen from the resulting clusterings, LSDBC provides an adequate separation of the original images, which makes it useful for the task of filtering trees [7] (see Figure 5).

6 Related Work and Density Estimation

In Ester *et al.* [1], density based clustering (DBSCAN) was presented as a clustering technique which can discover clusters of arbitrary shape. Hinneburg and Keim [8] introduced a new density based clustering technique as DENCLUE, which sums the density impact of a data point within its neighborhood. In effect, density based clustering methods estimate the density of points in a given dataset to cluster and differentiate densely populated regions. Two methods are commonly used for *density estimation*: Parzen windows and k-nearest neighbor (*kNN*) estimation [9]. In Parzen windows, we assume the existence of a d dimensional hypercube with volume $V = \varepsilon^d$, where ε is the length of an edge of the hypercube. Then, the number of points falling within this volume gives an estimate of the density (pick a radius and count the number of neighbors). Problems arise when choosing ε, which determines the volume, V, also known as the problem of finding the best window size. In *kNN*, we choose k nearest neighbors and grow the corresponding ε and volume V until it encloses the $k + 1$ points (pick a number of neighbors and compute the radius).

Density based clustering algorithms can also be divided into two types based on how they estimate the density: Parzen window type and *kNN* type. Among the Parzen window type approaches, we can count DBSCAN [1], DENCLUE [8], and CLIQUE [10]. Most of these algorithms suffer from the problem of choosing the best window size. LSDBC and OPTICS [2] are both *kNN* type density based clustering algorithms. OPTICS focuses on providing a representation of data (e.g. reachability plots) that enables different clusterings to be implemented and generates an ordering of points based on the order in which they are explored by the subroutine ExpandClusterOrder. LSDBC starts with a density based ordering and performs cluster expansions starting with the



DBSCAN, $r = 0.04$, density=5:
Only highly dense regions are recognized.

LSDBC, $k = 6,\ \alpha = 3$:
Background clutter is divided into 2 clusters.

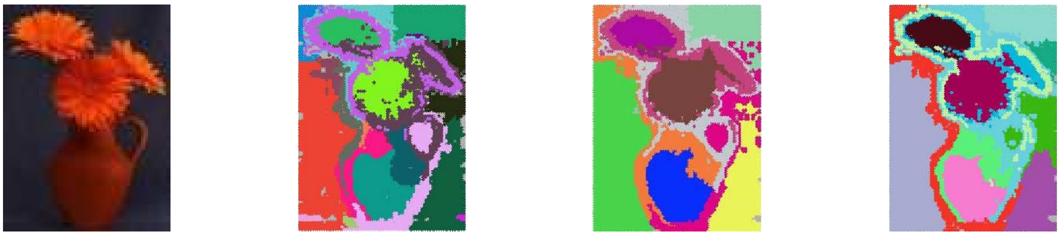
k-means, $k = 20$:
Densely populated regions are divided.

LSDBC, $k = 7,\ \alpha = 3$:
Background clutter is divided into 2 clusters.

Spectral with local scaling, $k = 20$:
Densely populated regions are divided,
diversely populated regions merged.

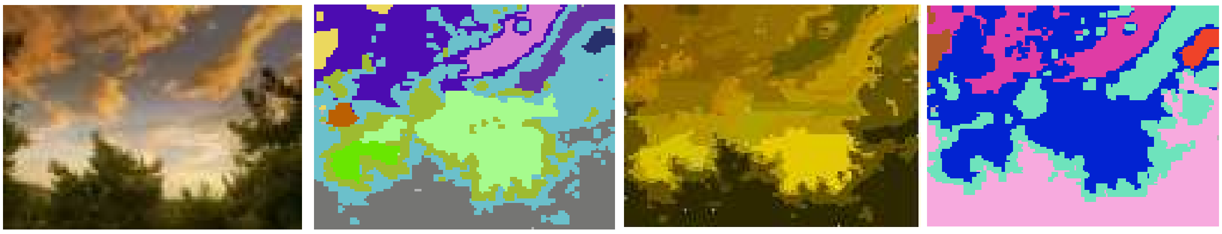
LSDBC, $k = 8,\ \alpha = 3$:
Background clutter is classified as a single
cluster.

Fig. 3. Comparison of clustering performance on a dataset with different local statistics.



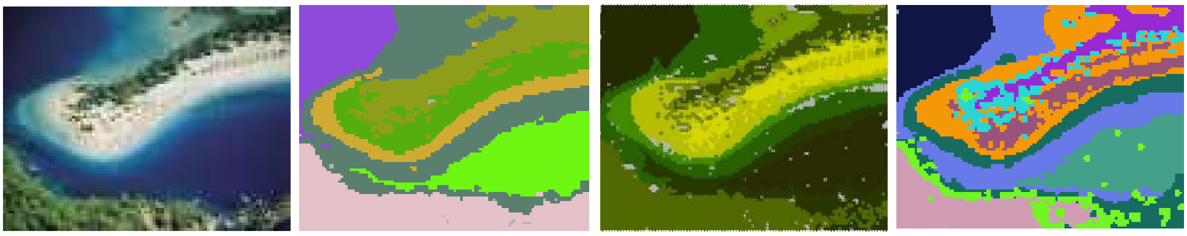
Flowers in a vase (original) LSDBC: $k = 7, \ \alpha = 6$ $k = 22, \ \alpha = 3$ $k = 13, \ \alpha = 5$

Fig. 4. Segmentation of a still life image. Notice the identification of the same transparent regions, which can be seen through the handle of the vase.



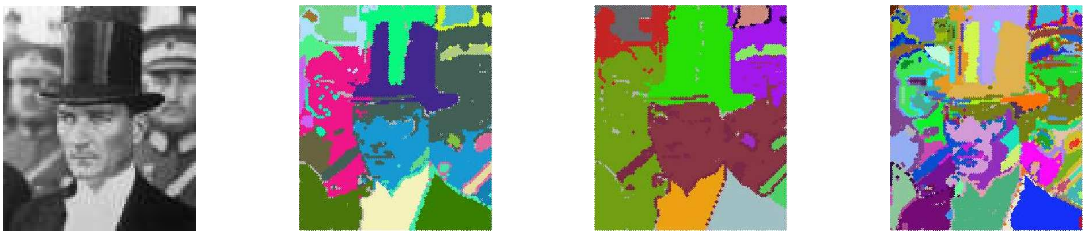
Trees (original) LSDBC: $k = 17, \ \alpha = 5$ $k = 7, \ \alpha = 4$ $k = 13, \ \alpha = 9$

Fig. 5. Segmentation of a group of trees and the sky.



Oludeniz (original) LSDBC: $k = 12, \ \alpha = 5$ $k = 10, \ \alpha = 6$ $k = 13, \ \alpha = 6$

Fig. 6. Segmentation of an image of a seaside, Oludeniz.



Ataturk (original) LSDBC: $k = 10, \ \alpha = 7$ $k = 10, \ \alpha = 8$ $k = 6, \ \alpha = 5$

Fig. 7. Segmentation of an image of Ataturk.

densest available point. Also, the cut-off for clusters in OPTICS is decided based on the density gradient of the edges of clusters, whereas LSDBC bases its cut-off on the density of the center of the cluster, which we believe to be more robust and noise free.

7 Conclusion

We have introduced the locally scaled density based clustering method. LSDBC discovers local maxima of density using a k-nearest-neighbor density estimation method and grows each cluster until the density falls below a pre-specified ratio of the center point's density. The resulting clustering technique is able to identify clusters of arbitrary shape on noisy backgrounds that contain significant density gradients. The performance of the new algorithm is demonstrated on a number of synthetic datasets and real images and shown to be promising for a broad range of its parameters.

LSDBC can be effectively used as a tool for summarizing the inherent relationships within the data. We have shown that LSDBC's performance in differentiating between densely populated regions is better than other clustering algorithms that we considered. LSDBC can also be used to summarize and segment images into meaningful regions.

References

1. Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, pages 226–231, 1996.
2. Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: ordering points to identify the clustering structure. In *SIGMOD '99: Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, pages 49–60, New York, NY, USA, 1999. ACM Press.
3. Lihi Zelnik-Manor and Pietro Perona. Self-tuning spectral clustering. In *Eighteenth Annual Conference on Neural Information Processing Systems*, 2004.
4. M. Emre Celebi, Y. Alp Aslandogan, and Paul R. Bergstresser. Mining biomedical images with density-based clustering. In *ITCC '05: Proceedings of the International Conference on Information Technology: Coding and Computing*, volume I, pages 163–168, Washington, DC, USA, 2005. IEEE Computer Society.
5. Jörg Sander, Martin Ester, Hans-Peter Kriegel, and Xiaowei Xu. Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data Mining and Knowledge Discovery*, 2(2):169–194, 1998.
6. Pietro Perona and William T. Freeman. A factorization approach to grouping. In *ECCV '98: Proceedings of the 5th European Conference on Computer Vision*, volume I, pages 655–670, London, UK, 1998. Springer-Verlag.
7. Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: an efficient data clustering method for very large databases. *SIGMOD Record*, 25(2):103–114, 1996.
8. Alexander Hinneburg and Daniel A. Keim. An efficient approach to clustering in large multimedia databases with noise. In *KDD*, pages 58–65, 1998.
9. Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.
10. Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *SIGMOD '98: Proceedings ACM SIGMOD International Conference on Management of Data, June 2-4, 1998, Seattle, Washington, USA*, pages 94–105. ACM Press, 1998.