Summary: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Martin Ester, Hans-Peter Kriegel, Jorg Sander, Xiaowei Xu

Main point: The authors introduce a clustering algorithm based on the notion of density, where clusters are considered as the sets of points that lie inside or on the border of high-density regions in spatial databases. They show that it performs well on large spatial databases with clusters of arbitrary shapes and scales almost linearly with the number of points.

The authors begin by examining the deficiencies of existing partitioning and hierarchical algorithms. The argue that because the clusters discovered by partitioning algorithms such as $k$-means or $k$-medoids require $k$ as an input, and because these algorithms can only find clusters that are convex, the algorithms are restrictive. Ng and Han’s CLARANS (1994) algorithm is discussed as the state-of-the-art in large-scale clustering. Ester et al. state that although CLARANS is more efficient than $k$-means or $k$-medoids, it requires $O(n)$ runs in order to determine the “natural” number $k_{\text{natural}}$ of the number of clusters, and therefore becomes inefficient when $k$ is unknown. Furthermore, CLARANS assumes that “all objects to be clustered reside in main memory...which does not hold for large databases.” Hierarchical algorithms suffer from the difficulty of choosing appropriate termination criteria, which determine when either a) in the case of an agglomerative algorithm, the algorithm stops merging clusters, or b) in a divisive approach, the condition when no more clusters are split. The EJcluster algorithm, a hierarchical algorithm that was recently proposed at the time of Ester et al.’s publication, successfully discovers clusters of non-convex shape and automatically derives a stopping condition. However, its runtime is $O(n^2)$, making it prohibitively expensive for large databases. Jain proposed a density-based approach to clustering in 1988, which finds clusters in $k$-dimensional datasets, but the approach relies on the construction of multidimensional histograms which can be used to identify clusters, again a prohibitively expensive procedure.

In section 3, the authors propose their notion of “density-based” clusters. The attempt to formalize the idea that clusters are sets of points that belong to very dense regions, and that the density of points in noisy areas is significantly lower than the density within cluster regions. The key idea is that “for each point of a cluster the neighborhood of a given radius has to contain at least a minimum number of points, i.e. the density in the neighborhood has to exceed some threshold. The authors distinguish between “core points”, those in very dense regions of the cluster, and “border points”, those on the boundary of the cluster and whose neighborhood will not contain as many points as the neighborhood of core points. An example of a core point $q$ and a border point $p$ is illustrated in Figure 1. Therefore, in the clustering algorithm presented, the authors will require that for every point $p$ in a cluster $C$, there must be a point $q \in C$ such that $p$ is in the $\epsilon$-neighborhood of $q$, in addition to the requirement that $q$ must have a minimum number of points $\text{MinPts}$ in its $\epsilon$-neighborhood.

The authors formalize these requirement in the following way: A point $p$ is
directly density-reachable from a point $q$ if $p \in N_\epsilon(q)$ and $|N_\epsilon(q)| \geq \text{MinPts}$, where $N_\epsilon(q)$ is the $\epsilon$-neighborhood of $q$. A point $p$ is density-reachable from $q$ if there is a chain of points $q, p_1, \ldots, p_m, p$ such that $p_{i+1}$ is density-reachable from $p_i$, $p_1$ is density-reachable from $q$, and $p$ is density-reachable from $p_m$. The authors note that density-reachability is transitive but not symmetric unless both $p$ and $q$ are core points. To include two border points in the same cluster, the authors require the points to be density-connected, defined as follows: A point $p$ is density-connected to a point $q$ wrt $\epsilon$ and $\text{MinPts}$ if there is a point $o$ such that both $p$ and $q$ are density-reachable from $o$ wrt $\epsilon$ and $\text{MinPts}$. Note that density-connectivity is a symmetric relation. Using these definitions, the authors define a cluster in the following way:

**Definition 1** Let $D$ be a database of points. A cluster $C$ wrt $\epsilon$ and $\text{MinPts}$ is a non-empty subset of points satisfying the following conditions:
1) $\forall p, q: \text{if } p \in C \text{ and } q \text{ is density-reachable from } p \text{ wrt } \epsilon \text{ and } \text{MinPts}, \text{ then } q \in C$
2) $\forall p, q \in C: p \text{ is density-connected to } q \text{ wrt } \epsilon \text{ and } \text{MinPts}$

Noise is defined simply as the set of points in $D$ not belonging to any of its clusters: If $C_1, \ldots, C_k$ are $k$ clusters in a dataset $D$, then noise is the set of points $p \in D$ such that for all $i$, $p \notin C_i$ wrt parameters $\epsilon$ and $\text{MinPts}$.

The authors state Lemmas that show that any cluster $C$ by their definition contains exactly the points which are density-reachable from an arbitrary core point in $C$. This fact is important for the development of their algorithm, which they call DBSCAN.

DBSCAN proceeds as follows: start with an arbitrary point $p$ and find all points that are density-reachable from $p$. If $p$ is a border point, no points will
be density-reachable from it, and another arbitrary point \( \hat{p} \) is picked at random from the database. If \( p \) is a core point, then by the Lemmas stated, finding all the points density-reachable from \( p \) is equivalent to finding a cluster in the database. Since DBSCAN may merge two “true” clusters if the parameters \( \text{MinPts} \) and \( \epsilon \) are chosen to be too small and too large, respectively, a recursive application of DBSCAN yields a finer-grained clustering of the dataset and can be applied as many times as desired. In order to find all points that are density-reachable from \( p \), the algorithm proceeds in a breadth-first-search fashion, exploring the \( \epsilon \)-neighborhood of \( p \) initially, then the neighborhoods of the \( \epsilon \)-neighbors of \( p \) in sufficiently dense regions, and so on. Points are marked as noise if they are not in a sufficiently dense region, and may be identified as members of a cluster in a later iteration if they are density-reachable from at least one point. An R*-tree is used to search the neighborhood of a given point.

To determine the \( \epsilon \) and \( \text{MinPts} \) parameters, a heuristic is employed. All the points in the database are ordered according to their distance to their \( k \)-th nearest neighbor, where in practice \( k = 4 \) is used. Then, \( \epsilon \) is chosen in a somewhat ad-hoc manner by picking the distance at which the first major increase in distance to the 4th nearest neighbor occurs. Intuitively this is the point separating noise from the points belonging to clusters, because noisy points are expected to lie in low-density regions.

Experiments are performed on synthetic data, where DBSCAN outperforms CLARANS and finds clusters more accurately, although the results were evaluated “by inspection” by the authors. Results are also shown for the SEQUOIA benchmark dataset, which is composed of “data sets that are representative of Earth Science tasks.” These results show that DBSCAN outperforms CLARANS in runtime by a factor that lies between 250 and 1900, and that DBSCAN exhibits scaling that is “slightly higher than linear” while CLARANS scales quadratically.