Figure 1: An example of a data set where it might be difficult to identify all the clusters $A, B, C$ and $D$, because density-based algorithms may merge $A$ and $B$ together or miss $C$, and algorithms based on the minimal distance-to-center objective may miss cluster $D$ due to its non-spherical structure. Here the points outside the dashed boundary are noise and do not belong to any cluster.

Summary: OPTICS: Ordering Points to Identify the Clustering Structure, by Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, Jorg Sander (all from Institute for Computer Science, University of Munich) (1999)

Main Point: The authors present an algorithm for ordering points in a database with respect to the density of the clusters within the data. In this way, clusters of differing densities and structures may be discovered.

The idea to order data points with respect to their density regions is motivated by the problem that many existing algorithms had at the time – they were either too inefficient, could only find convex clusters of similar size, couldn’t find clusters of different density, or suffered from mistaken joins of clusters (e.g., single linkage algorithm). For example, consider Figure 1. Here, we have clusters $A, B, C$ and $D$ which differ in both density and structure. Popular algorithms such as $k$-means or DBSCAN will have trouble discovering all four clusters because depending on the parameter settings, they may fail to find one or more cluster, or merge two together that should not be merged.

To fix this problem, the authors build on the idea behind the DBSCAN algorithm, where clusters are found based on the density of the $\epsilon$-neighborhoods of the data points (please see my summary of DBSCAN). The major issue with DBSCAN is that the resulting clustering depends on the choice of parameter settings $\epsilon$, defining the size of the neighborhood to be scanned, and $\text{MinPts}$, the
threshold density surrounding a point for it to be considered part of a cluster. The authors point out that high-density clusters by the DBSCAN definition will be completely contained within lower-density clusters. In other words, keeping MinPts constant, the clusters found by setting \( \epsilon \) to \( \epsilon_1 \) will be completely contained in the clusters found by setting \( \epsilon \) to \( \epsilon_2 > \epsilon_1 \). Thus, the authors propose finding such a hierarchical clustering simultaneously. To produce consistent results, they need to ensure that the data points are processed in order by highest density region to lowest.

To achieve this goal, the authors introduce two new definitions which extend those used by DBSCAN. These definitions are those of “core-distance” and “reachability-distance of an object \( p \) with respect to an object \( o \)”. The core-distance of an object \( p \) is the smallest distance \( \epsilon' \) between \( p \) and an object in its \( \epsilon \)-neighborhood such that \( p \) would be a core object wrt \( \epsilon' \) if this neighbor is contained in the \( \epsilon \)-neighborhood of \( p \). The reachability-distance of \( p \) with respect to another object \( o \) is the smallest distance such that \( p \) is directly density-reachable from \( o \) if \( o \) is a core object. Note that this depends on which object \( o \) is chosen for comparison with \( p \). The core-distance and reachability-distance are used by the OPTICS algorithm to order a database according to the density of regions surrounding points, thereby providing enough information to extract all density-based clusterings for \( \epsilon' \) smaller than or equal to the generating \( \epsilon \).

The run-time of OPTICS is \( O(n \times \text{run-time of an } \epsilon \text{-neighborhood query}) \). With a tree-based spatial index used for the \( \epsilon \)-neighborhood query, such as an R*-tree, X-tree, or M-tree, this run-time is \( O(\log n) \), thus the total running time of OPTICS is \( O(n \log n) \). To generate a density-based clustering with parameter \( \epsilon' \), the ordered database is simply scanned, and points are assigned to clusters based on their core-distance and reachability distance. The authors also present one way to analyze the density-ordered data by generating reachability plots to understand the underlying structure of the data.

The authors point out that “for very high-dimensional spaces, no index structures exist to efficiently support the hypersphere range queries needed by the OPTICS algorithm,” therefore it is an interesting future work direction to improve OPTICS to deal with this setting, either by efficient queries in high dimensions or to examine the trade-off between expensive queries and clustering quality.