Density-Based Clustering. Preliminaries

Density-based clustering algorithms is a family of algorithms that determine density-based clusters in the data. A formal definition of a density-based cluster is supplied below.

\( \varepsilon \)-neighborhood. Let \( D = \{ d_1, \ldots, d_n \} \) be a set of data points, and let \( \text{dist}() \) be a distance function for points in \( D \).

Given a number \( \varepsilon \), an \( \varepsilon \)-neighborhood point \( d \in D \) is defined as:

\[
N_\varepsilon(d) = \{ d_i \in D | d_i \neq d, \text{dist}(d, d_i) \leq \varepsilon \}
\]

Core points. Given an integer \( \text{minpts} > 0 \), a point \( d \in D \) is a core point in \( D \) if

\[
|N_\varepsilon(d)| \geq \text{minpts},
\]

that is, if the \( \varepsilon \)-neighborhood of \( d \) contains \( \text{minpts} \) or more points.

Border (boundary) points. A point \( d \in D \) is a border (boundary) point if

\[
|N_\varepsilon(d)| < \text{minpts},
\]

but

\[
(\exists d' \in D)(d \in N_\varepsilon(d')),
\]

i.e., if the \( \varepsilon \)-neighborhood of \( d \) contains fewer than \( \text{minpts} \) points, but \( d \) itself is in a \( \varepsilon \)-neighborhood of some other point \( d' \in D \).

\(^1\) A similar definition will also work for a similarity function.
Noise points. A point \( d \in D \) is a noise point if it is neither core point nor boundary point in \( D \).

Density-reachability. Given the density radius \( \varepsilon \) and the minimum density \( \text{minpts} \), a point \( d' \in D \) is directly density-reachable from point \( d \in D \) if \( d' \in N_\varepsilon(d) \). \( d' \) is density-reachable from \( d \) if there exists a chain of points \( d = d_1, d_2, \ldots, d_k = d' \), such that \( d_i \in N_\varepsilon(d_{i-1}) \).

Note: Density-reachability is an asymmetric relationship (a boundary point \( x \) may be density-reachable from a core point \( y \), but not the other way around).

Density connectivity. Two points \( d \in D \) and \( d' \in D \) are density connected, if there exists a core point \( f \in D \), such that both \( d \) and \( d' \) are density-reachable from \( f \).

Density-based cluster. A density cluster \( D' \subset D \) is any maximal set of points that are density-connected to each other.

DBSCAN

DBSCAN is a key algorithm for discovery of density-based clusters. DBSCAN takes as input a dataset \( D \), a distance function \( \text{dist}() \) that is defined on all pairs of points from \( D^2 \) and two parameters:

- \( \varepsilon \): the radius of the \( \varepsilon \)-neighborhood in which DBSCAN will search for data points;
- \( \text{minpts} \): the smallest number of points in a \( \varepsilon \)-neighborhood of a point, for it to be declared a core point.

The pseudocode for DBSCAN is shown in Figure 1.

The algorithm works as follows:

- **Core point discovery.** First, DBSCAN scans through the entire dataset \( d \) and determines based on \( \varepsilon \) and \( \text{minpts} \) parameters, the list of core points.

- **Cluster construction.** Each cluster is constructed as follows. The algorithm pulls a yet-to-be visited core point, and recursively computes all density connected points to it. It then proceeds to search for the next unvisited/unlabeled core point until it runs out of core points to expand.

- **Output.** At the end, the algorithm returns the breakdown of points into clusters, as well as the lists of core, boundary and noise points.

\(^2\text{Usually, DBSCAN uses Euclidean distance, but it can also use other distance functions. Also, a version of DBSCAN that uses similarity measures rather than distance measures, can be obtained from the pseudocode shown in these notes in a straightforward way.}\)
Algorithm DBSCAN\((D, \text{dist}(), \varepsilon, \text{minpts})\)

\begin{verbatim}
begin
Core := \emptyset;
for each \(d_i \in D\) do // find core points
    Compute \(N_\varepsilon(d_i)\);
    cluster\((d_i)\) := \emptyset; // initialize cluster assignment for the point
    if \(|N_\varepsilon(d_i)| \geq \text{minpts}\) then Core := Core \cup \{d_i\};
end for

CurrentCluster := 0; // initialize current cluster label
for each \(d \in Core\) do
    if cluster\((d) = \emptyset\) then
        CurrentCluster := CurrentCluster + 1; // start a new cluster
        cluster\((d) := CurrentCluster; // assign first point to the cluster
        DensityConnected\((D, d, Core, CurrentCluster)\); // find all density connected points
    endif
end for

ClusterList := \emptyset
for \(k := 1\) to \(CurrentCluster\) do // assemble clusters
    Cluster\([k]\) = \{d \in D|cluster\((d) = k\);\}
    ClusterList := ClusterList \cup Cluster[k];
end for
Noise := \{d \in D|cluster\((d) = \emptyset\}\)
Border := D − (Noise \cup Core)
return ClusterList, Core, Border, Noise
end
\end{verbatim}

function DensityConnected\((D, point, Core, clusterId)\)

\begin{verbatim}
begin
for each \(d \in N_\varepsilon(point)\) do // add all neighbors to cluster
    cluster\((d) := clusterId;\)
    if \(d \in Core\) then DensityConnected\((D, d, Core, clusterId)\);
    // recursively do it for each core point discovered
endfor
end
\end{verbatim}

Figure 1: Pseudocode for DBSCAN algorithm