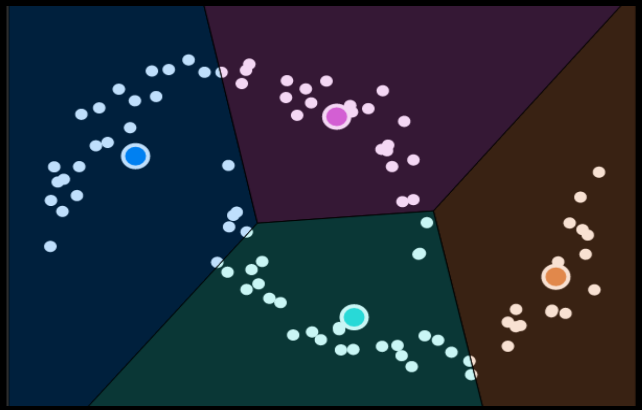
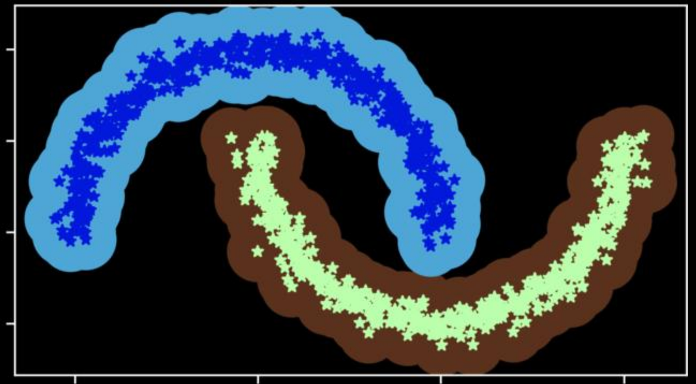
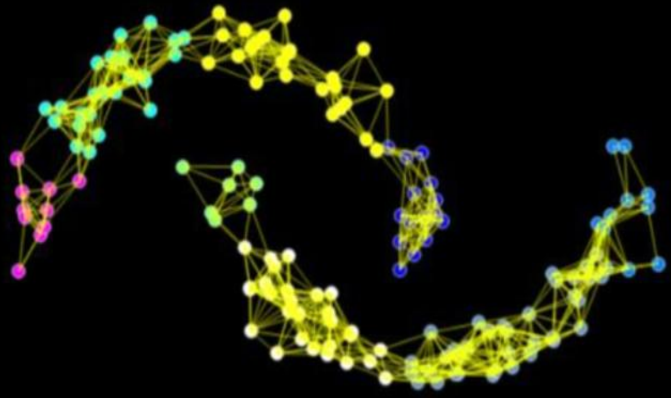
CONNECTING THE DOTS – DENSITY-CONNECTIVITY DISTANCE UNIFIES DBSCAN, K-CENTER AND SPECTRAL CLUSTERING

Anna Beer, Andrew Draganov, Ellen Hohma, Philipp Jahn, Christian Frey, Ira Assent

Presentation at KDD 2023
August 9th, 2023

AARHUS UNIVERSITY
LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN
Munich Center for Machine Learning
Fraunhofer IIS

RECAP: CLUSTERING

Centroid-based	Density-based	Spectral
		
k-Means, k-Center, …	DBSCAN, HDBSCAN, …	Spectral Clustering, SCAR, …

- Procedurally defined
- No loss function

[0] https://antoinebrl.github.io/blog/kmeans/ Last Accessed: 07.07.2023
[1] https://www.mygreatlearning.com/blog/introduction-to-spectral-clustering/ Last Accessed: 07.07.2023
[2] https://medium.com/@tpreethi/dbscan-algorithm-density-based-spatial-clustering-of-application-with-noise-a826538dcb42 Last Accessed: 07.07.2023

RESULT I

- DBSCAN is defined procedurally ☹
- We now know the loss function of DBSCAN

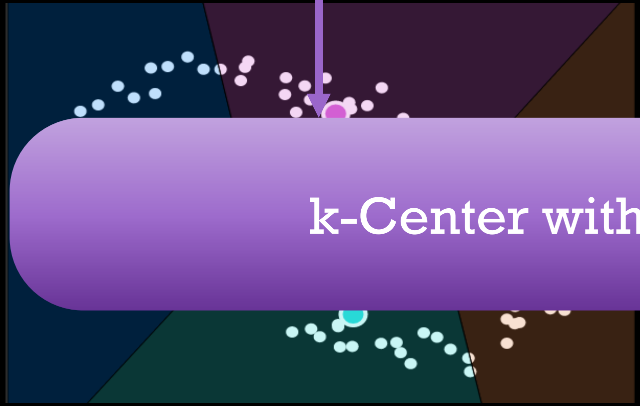
```
ExpandCluster(SetOfPoints, Point, ClId, Eps,
              MinPts) : Boolean;
  seeds:=SetOfPoints.regionQuery(Point,Eps);
  IF seeds.size<MinPts THEN // no core point
    SetOfPoint.changeClId(Point,NOISE);
    RETURN False;
  ELSE    // all points in seeds are density-
          // reachable from Point
    SetOfPoints.changeClIds(seeds,ClId);
    seeds.delete(Point);
    WHILE seeds <> Empty DO
      currentP := seeds.first();
      result := SetOfPoints.regionQuery(currentP,
                                         Eps);
      IF result.size >= MinPts THEN
        FOR i FROM 1 TO result.size DO
          resultP := result.get(i);
          IF resultP.ClId
              IN {UNCLASSIFIED, NOISE} THEN
               resultP.ClId = UNCLASSIFIED THEN
              seeds.append(resultP);
              D IF;
              tOfPoints.changeClId(resultP,ClId);
              IF; // UNCLASSIFIED or NOISE
              R;
              // result.size >= MinPts
              lete(currentP);
    END WHILE; // seeds <> Empty
    RETURN True;
  END IF
END; // ExpandCluster
```

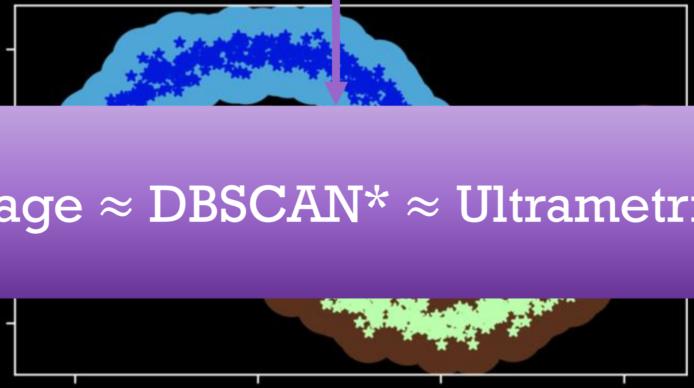
```
DBSCAN (SetOfPoints, Eps, MinPts)

// SetOfPoints is UNCLASSIFIED
  ClusterId := nextId(NOISE);
  FOR i FROM 1 TO SetOfPoints.size DO
    Point := SetOfPoints.get(i);
    IF Point.ClId = UNCLAS
      IF ExpandCluster(Set
             ClusterId, Ep
        ClusterId := nextI
      END IF
    END IF
  END FOR
END; // DBSCAN
```

$$\min_{\mathcal{C}\subset C} |\mathcal{C}| \quad d_{dc}(p,q) \le \varepsilon \, \forall p,q \in C_i \, \forall C_i \in \mathcal{C}$$

RESULT II

Density-Connectivity Distance d_{dc}^{μ}



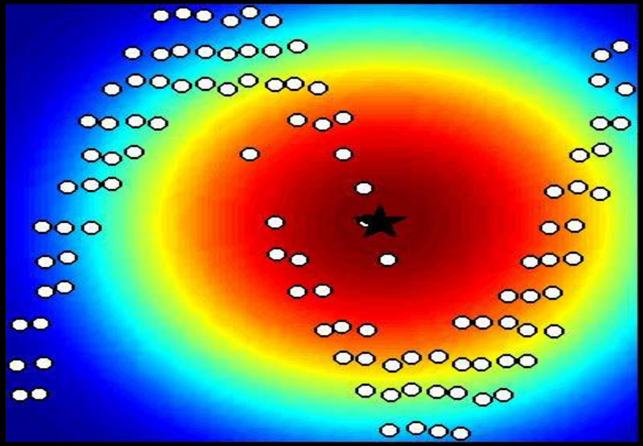


k-Center with q-coverage ≈ DBSCAN* ≈ Ultrametric Spectral Clustering

Density-Connectivity Distance d_{dc}^{μ}

A distance measure based on density-connectivity:

- Idea: Objects within a density-based cluster should be close, objects in different clusters are far apart
- Euclidean distance is not suitable for density-based clusters like "two moons"



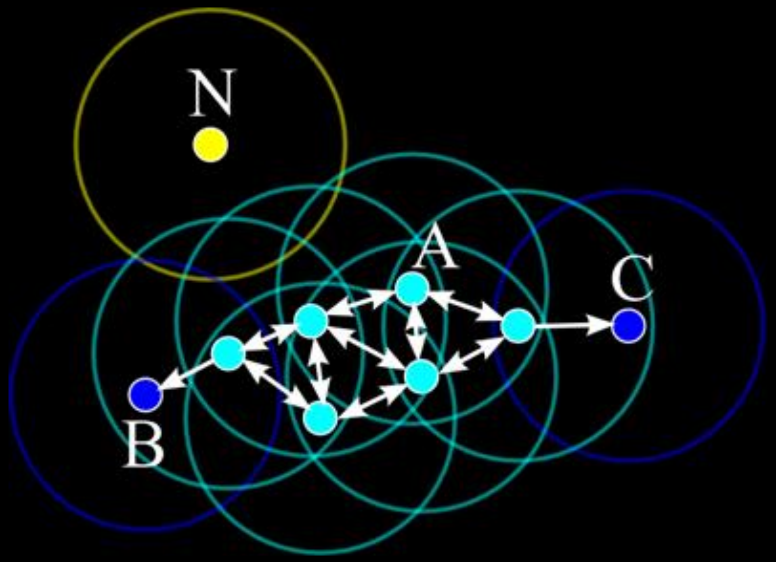
Euclidean distances from star-shaped point to other points

Kim, K. H., & Choi, S. (2013, June). Walking on minimax paths for k-nn search. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 27, No. 1, pp. 518-525).

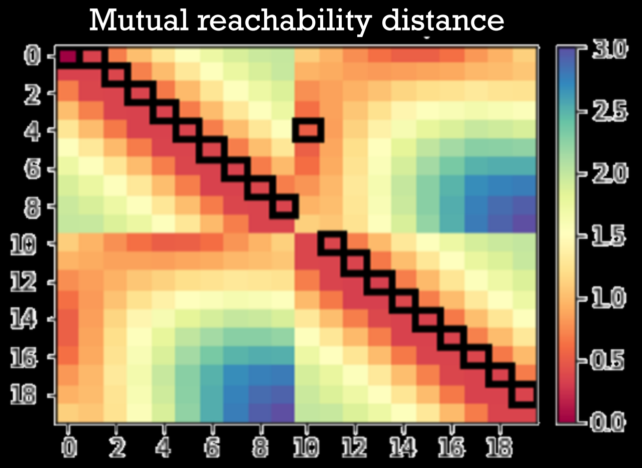
BACKGROUND: DENSITY-CONNECTIVITY

- Mostly known from DBSCAN
- Concept even older: Density-based hierarchical clustering approach from Wishart, 1969
- Points are *core points* or *dense* if they have more than μ neighbors in its ε-range
- Points are *density-connected* if there is a chain of dense points between them with links all shorter than ε
- *Mutual reachability distance* (similar versions also known from OPTICS, LOF):

$$d_r^\mu(p,q) = \max(d_{euclidean}(p,q), d_{core}^\mu(p), d_{core}^\mu(q))$$



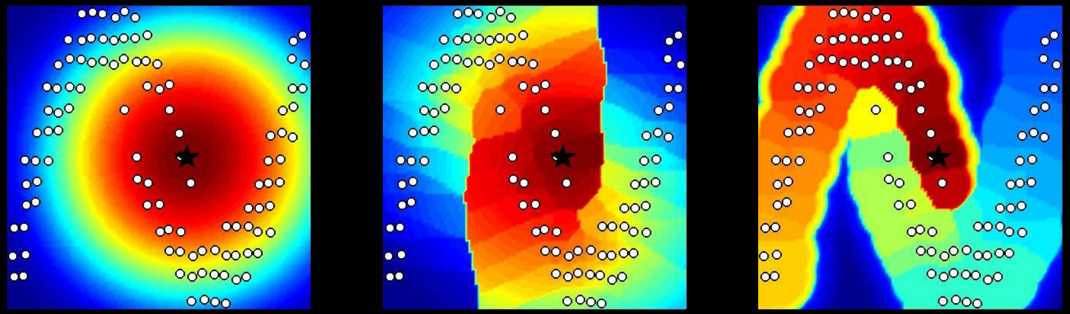
Density-connectivity, light blue points are core points [4]



[4] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. 1999. OPTICS: Ordering Points to Identify the Clustering Structure. In Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data (Philadelphia, Pennsylvania, USA) (SIGMOD '99). Association for Computing Machinery, New York, NY, USA, 49–60

BACKGROUND: MINIMAX PATHS

- Aka widest path, bottleneck shortest path, longest leg path, transitive distance, or connectivity kernel
- *Minimax path*: path that **min**imizes the **max**imum weight of any of its edges
- Minimax paths always lie on minimum spanning trees (MST) of a graph
- Minimax (path) distance: $d_m^{\mu,\delta}(p,q) = \min_{P \in \mathcal{P}(p,q)} \max_{e \in P} w^{\delta}(e)$



(a) Euclidean (b) Shortest path (c) Minimax distance

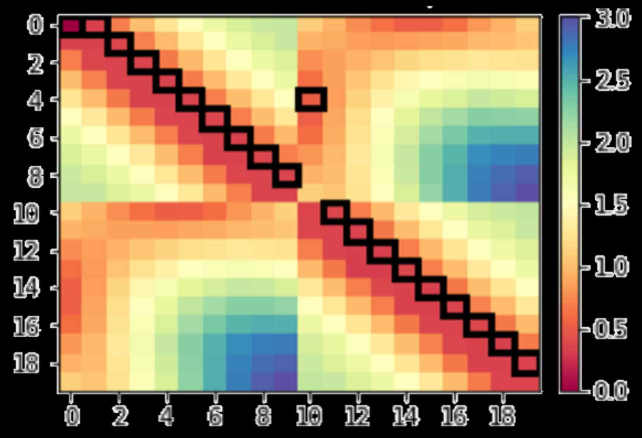
Distances from star-shaped point to other points

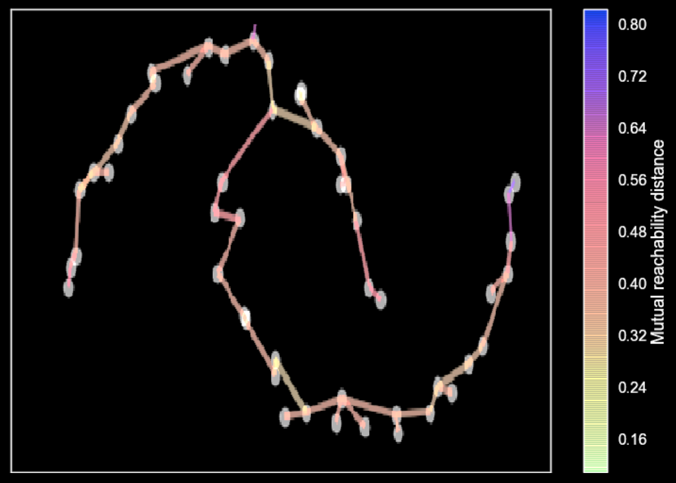
Kim, K. H., & Choi, S. (2013, June). Walking on minimax paths for k-nn search. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 27, No. 1, pp. 518-525).

NEW DISTANCE MEASURE: DC-DIST d_{dc}

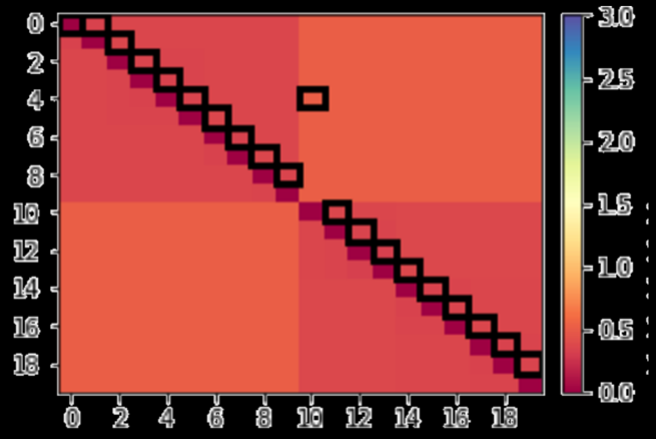
Mutual Reachability + Minimax distance = Density-connectivity distance

Mutual reachability distance





Density-connectivity distance



Minimum spanning tree of mutual reachability distance graph

L. McInnes, J. Healy, S. Astels, *hdbscan: Hierarchical density based clustering* In: Journal of Open Source Software, The Open Journal, volume 2, number 11. 2017
File: _images/how_hdbscan_works_10_1.png. Retrieved March 2, 2023 from https://hdbscan.readthedocs.io/en/latest/_images/how_hdbscan_works_10_1.png

NEW DISTANCE MEASURE: DC-DIST d_{dc}

Mutual Reachability + Minimax distance =

Definition 5: (cluster) Let D be a database of points. A *cluster* C wrt. Eps and MinPts is a non-empty subset of D satisfying the following conditions:
1) $\forall$ p, q: if p $\in$ C and q is density-reachable from p wrt. Eps and MinPts, then q $\in$ C. (Maximality)
2) $\forall$ p, q $\in$ C: p is density-connected to q wrt. EPS and MinPts. (Connectivity)

Our new distance measure „dc-distance“ is …

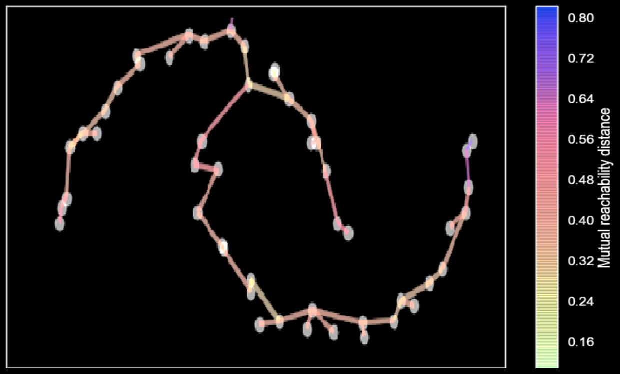
- **…the smallest ε, s.t. two points are core points of the same DBSCAN cluster**
- …computable with Kruskal‘s algorithm for building MSTs
- …an ultrametric (i.e., stronger Δ-inequality) with some cool properties.

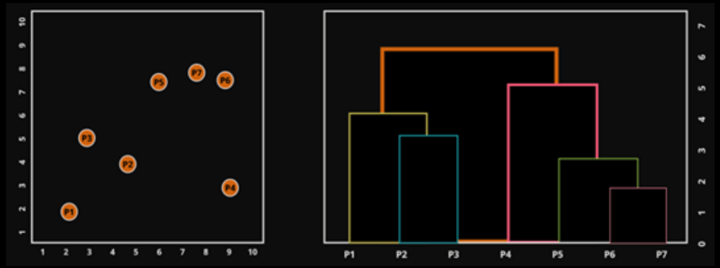
- … the basis for a loss function

$$\min_{\substack{\mathcal{C}\subset C \\ d_{dc}(p,q)\le\varepsilon\,\forall p,q\in C_i\,\forall C_i\in\mathcal{C}}} |\mathcal{C}|$$

NEW DISTANCE MEASURE: DC-DIST d_{dc}

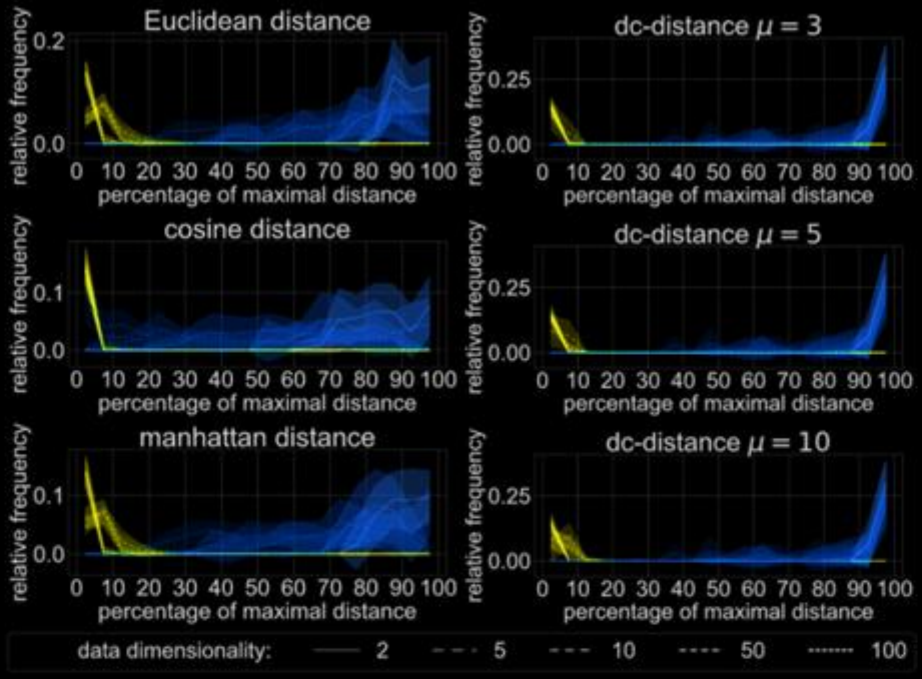
It‘s a tree! → Ultrametric



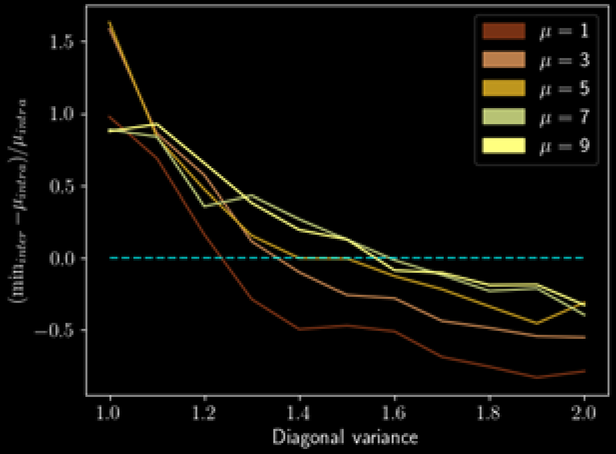


(**Sources:** L. McInnes, J. Healy, S. Astels, *hdbscan: Hierarchical density based clustering* In: Journal of Open Source Software, The Open Journal, volume 2, number 11. 2017
File: _images/how_hdbscan_works_10_1.png. Retrieved March 2, 2023 from https://hdbscan.readthedocs.io/en/latest/_images/how_hdbscan_works_10_1.png, https://i.stack.imgur.com/YjJfE.png)

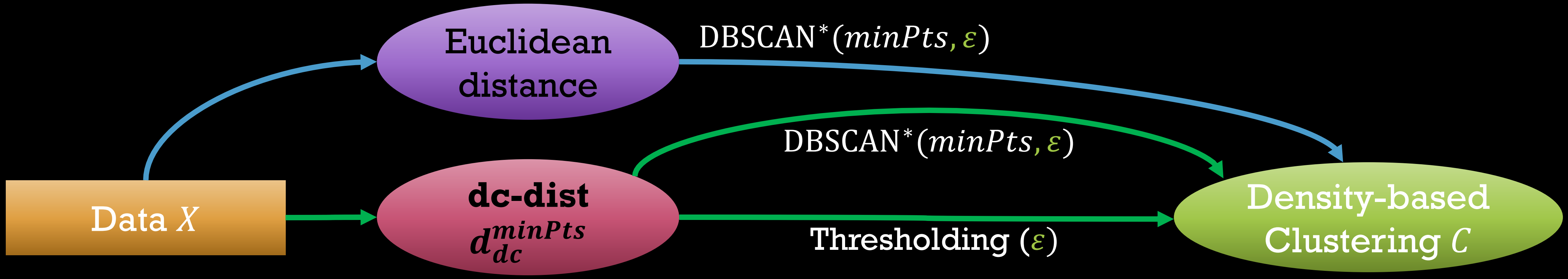
Clear distinction between intra- and inter-cluster distances



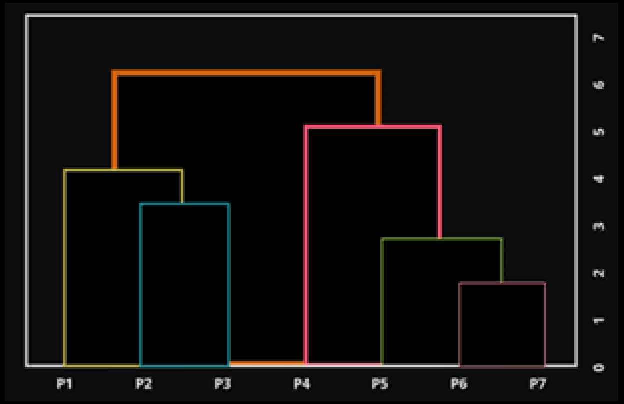
Density alleviates single-link effect



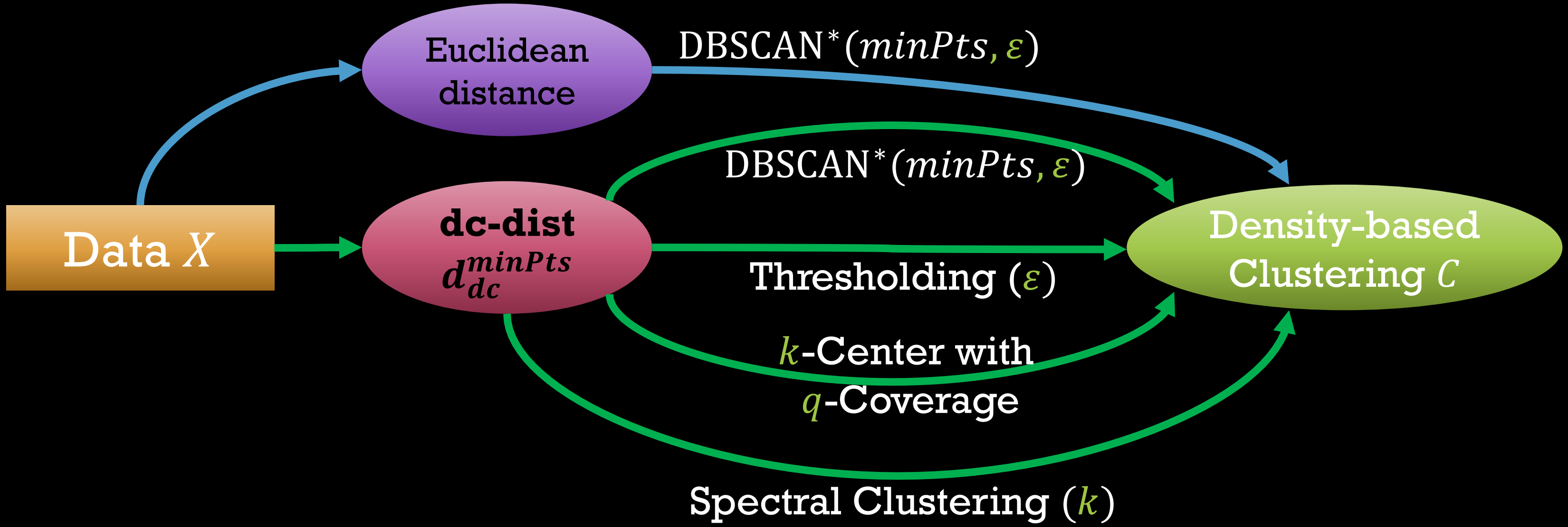
DC-DIST CAPTURES DENSITY-CONNECTIVITY



DC-DISTANCE IS AN ULTRAMETRIC (ROOTED TREE-METRIC)



Tree given by an ultrametric like dc-distance
(Source: https://i.stack.imgur.com/YjJfE.png)



Working on trees, several clustering methods lead to the same partitionings:

- DBSCAN* (dc-dist captures the relevant distances)
- Ultrametric Spectral Clustering finds the Mincut
- k-Center ***minimizes*** the ***maximum*** distance of any point to its cluster center
  - k-Center finds minimum ε s.t. there are k clusters
  - k-Center with q-Coverage has at least q points per cluster ➔ $q \sim minPts$

OVERVIEW DC-DIST