The Importance of Data Clustering Stability

Ezekiel Cochran

Advised By
Dr. Deborah Blevins - LeTourneau University
Dr. Robert Bridges - Oak Ridge National Labratory

LeTourneau University

February 8, 2024

What is Data Clustering?

- Data clustering is a technique used to group similar data points together.

What is Data Clustering?

- Data clustering is a technique used to group similar data points together.
- It helps to identify patterns and structures in large datasets.

What is Data Clustering?

- Data clustering is a technique used to group similar data points together.
- It helps to identify patterns and structures in large datasets.
- Clustering is useful for tasks such as
  - Pattern Recognition
  - Image Analysis
  - Machine Learning
  - Search Engines

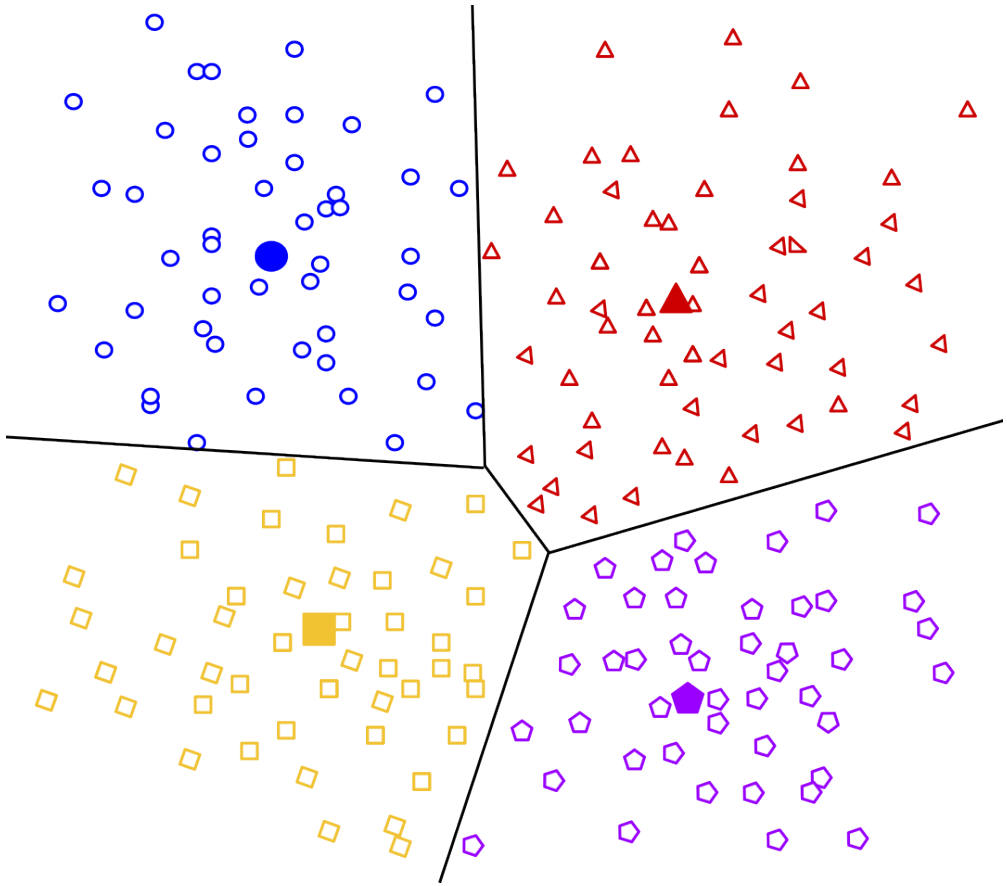
What is Data Clustering?

- Data clustering is a technique used to group similar data points together.
- It helps to identify patterns and structures in large datasets.
- Clustering is useful for tasks such as
  - Pattern Recognition
  - Image Analysis
  - Machine Learning
  - Search Engines
  - Sports Training?

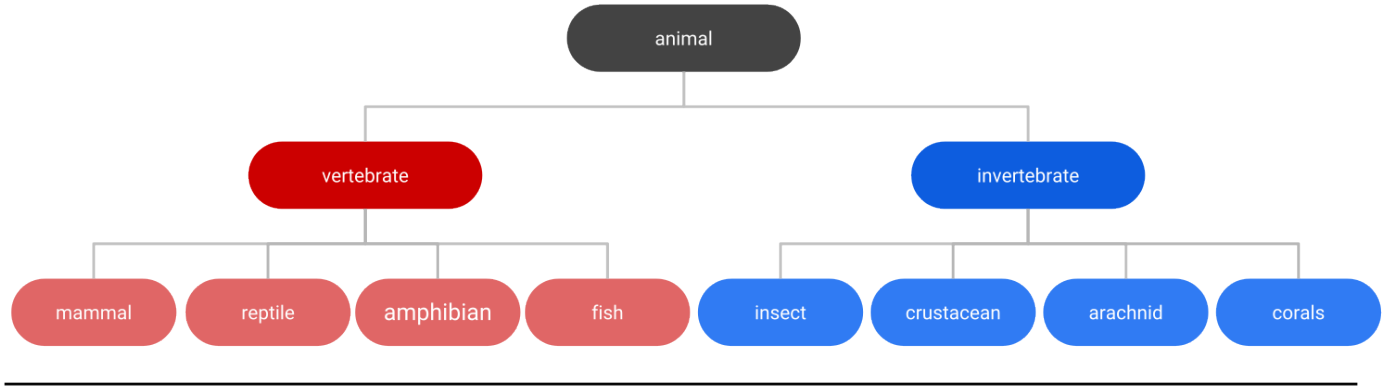
What is Data Clustering?

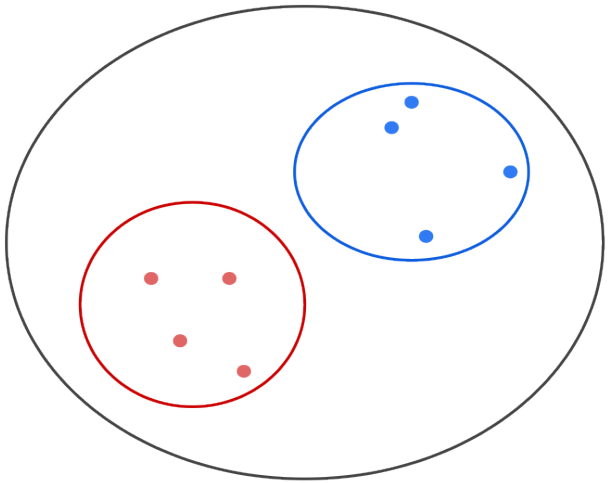
- Data clustering is a technique used to group similar data points together.
- It helps to identify patterns and structures in large datasets.
- Clustering is useful for tasks such as
  - Pattern Recognition
  - Image Analysis
  - Machine Learning
  - Search Engines
  - Sports Training?
- It can uncover hidden insights and make data more manageable and understandable.

Centroid-Based Clustering

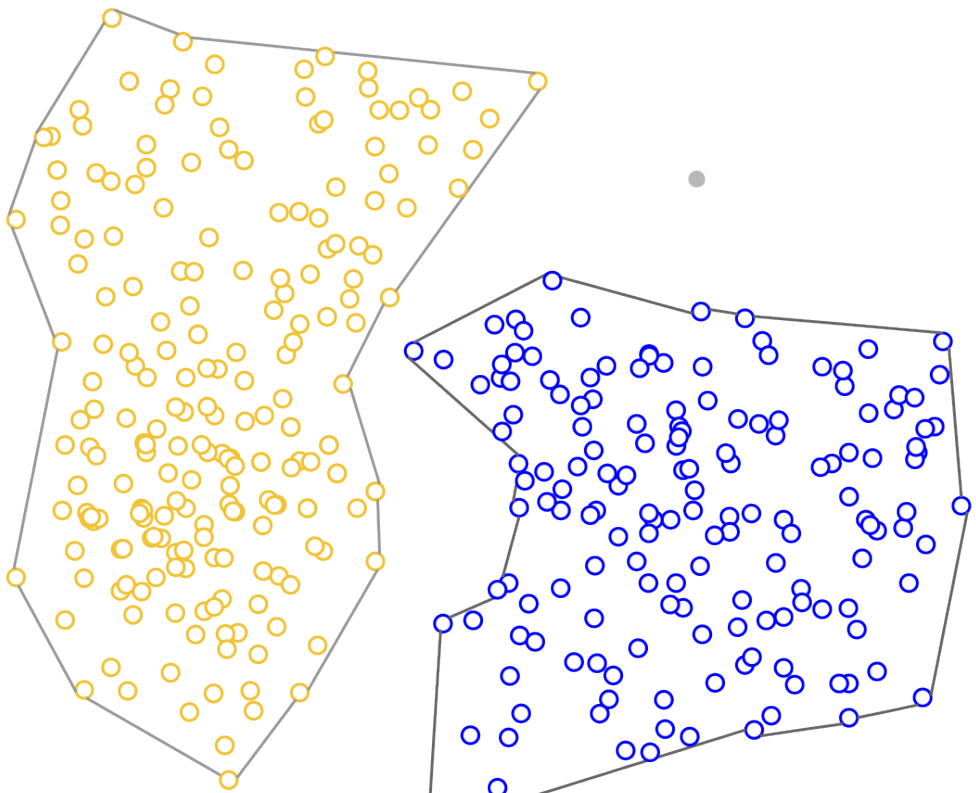


Hierarchical Clustering

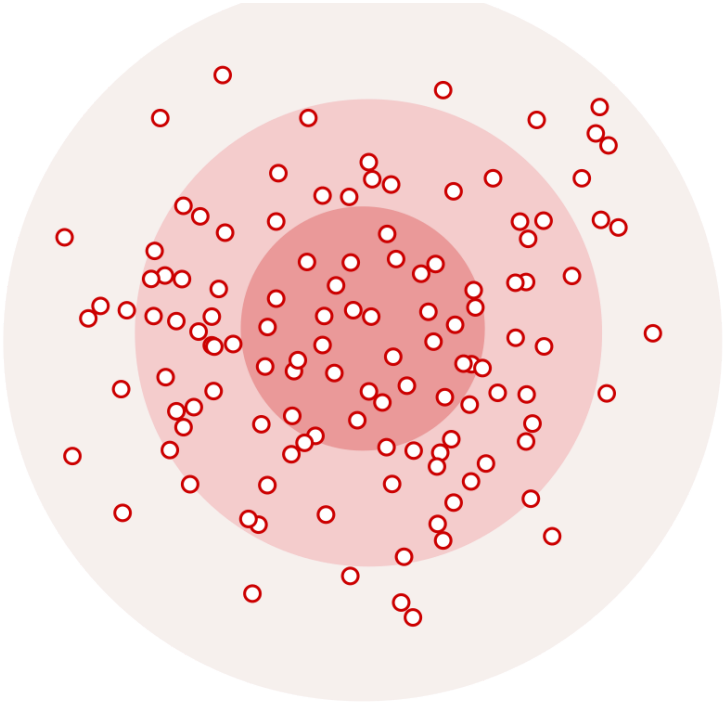


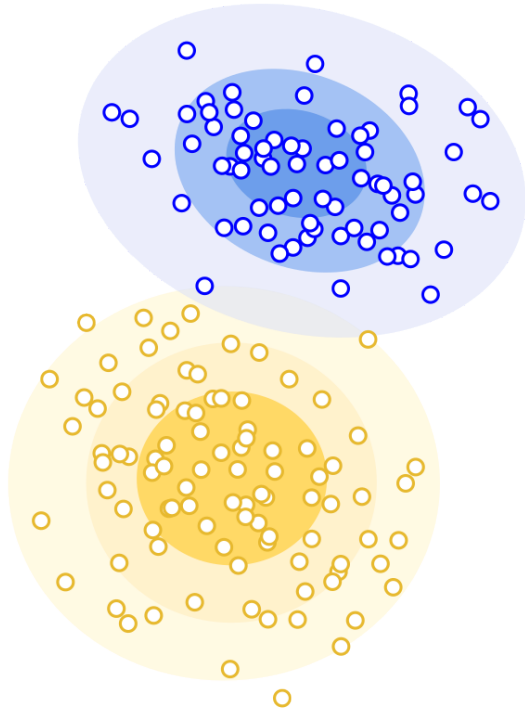


Density-Based Clustering



Distribution-Based Clustering





Common Algorithms

- K-Means
  - minimizes the sum of squared distances from each point to its assigned centroid

Common Algorithms

- K-Means
  - minimizes the sum of squared distances from each point to its assigned centroid
  - *requires the number of clusters to be specified*

Common Algorithms

- K-Means
  - minimizes the sum of squared distances from each point to its assigned centroid
  - *requires the number of clusters to be specified*
- Hierarchical Clustering

Common Algorithms

- K-Means
  - minimizes the sum of squared distances from each point to its assigned centroid
  - *requires the number of clusters to be specified*
- Hierarchical Clustering
  - Agglomerative: starts with each point as its own cluster, and merges the closest clusters

Common Algorithms

- K-Means
  - minimizes the sum of squared distances from each point to its assigned centroid
  - *requires the number of clusters to be specified*
- Hierarchical Clustering
  - Agglomerative: starts with each point as its own cluster, and merges the closest clusters
  - Divisive: starts with all points in one cluster, and repeatedly splits the cluster

Common Algorithms

- K-Means
  - minimizes the sum of squared distances from each point to its assigned centroid
  - *requires the number of clusters to be specified*
- Hierarchical Clustering
  - Agglomerative: starts with each point as its own cluster, and merges the closest clusters
  - Divisive: starts with all points in one cluster, and repeatedly splits the cluster
- DBSCAN
  - *Density-Based Spatial Clustering of Applications with Noise*

Common Algorithms

- K-Means
  - minimizes the sum of squared distances from each point to its assigned centroid
  - *requires the number of clusters to be specified*
- Hierarchical Clustering
  - Agglomerative: starts with each point as its own cluster, and merges the closest clusters
  - Divisive: starts with all points in one cluster, and repeatedly splits the cluster
- DBSCAN
  - *Density-Based Spatial Clustering of Applications with Noise*
- Gaussian Mixture Models
  - assumes that the data is generated from a mixture of several Gaussian distributions

- HDBSCAN stands for “Hierarchical Density-Based Spatial Clustering of Applications with Noise”

- HDBSCAN stands for "Hierarchical Density-Based Spatial Clustering of Applications with Noise"
- It was introduced in 2015, by Campello, Moulavi, and Zimek [2], for automatic clustering of n-dimensional data sets.

- HDBSCAN stands for "Hierarchical Density-Based Spatial Clustering of Applications with Noise"
- It was introduced in 2015, by Campello, Moulavi, and Zimek [2], for automatic clustering of n-dimensional data sets.
- It uses a "mutual reachability distance", and identifies clusters by iteratively merging or splitting groups of data points.

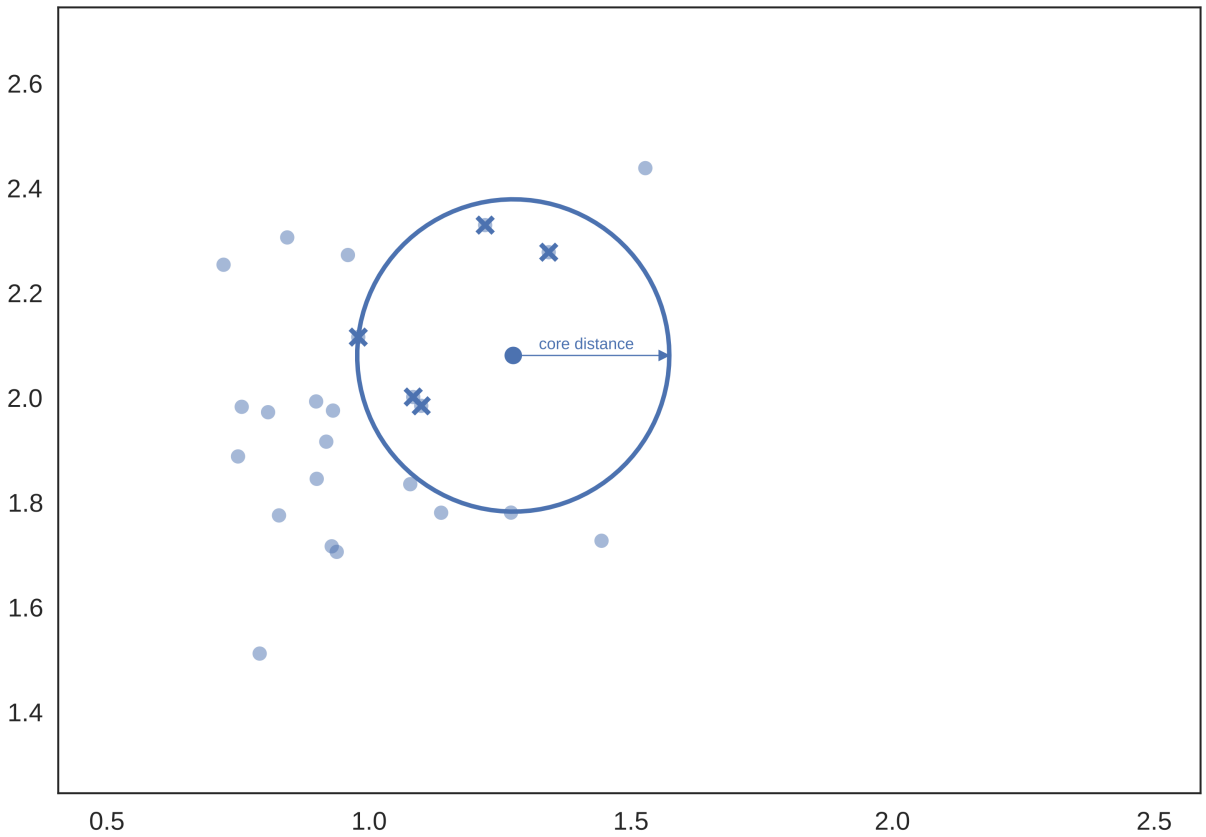
core_k

Definition

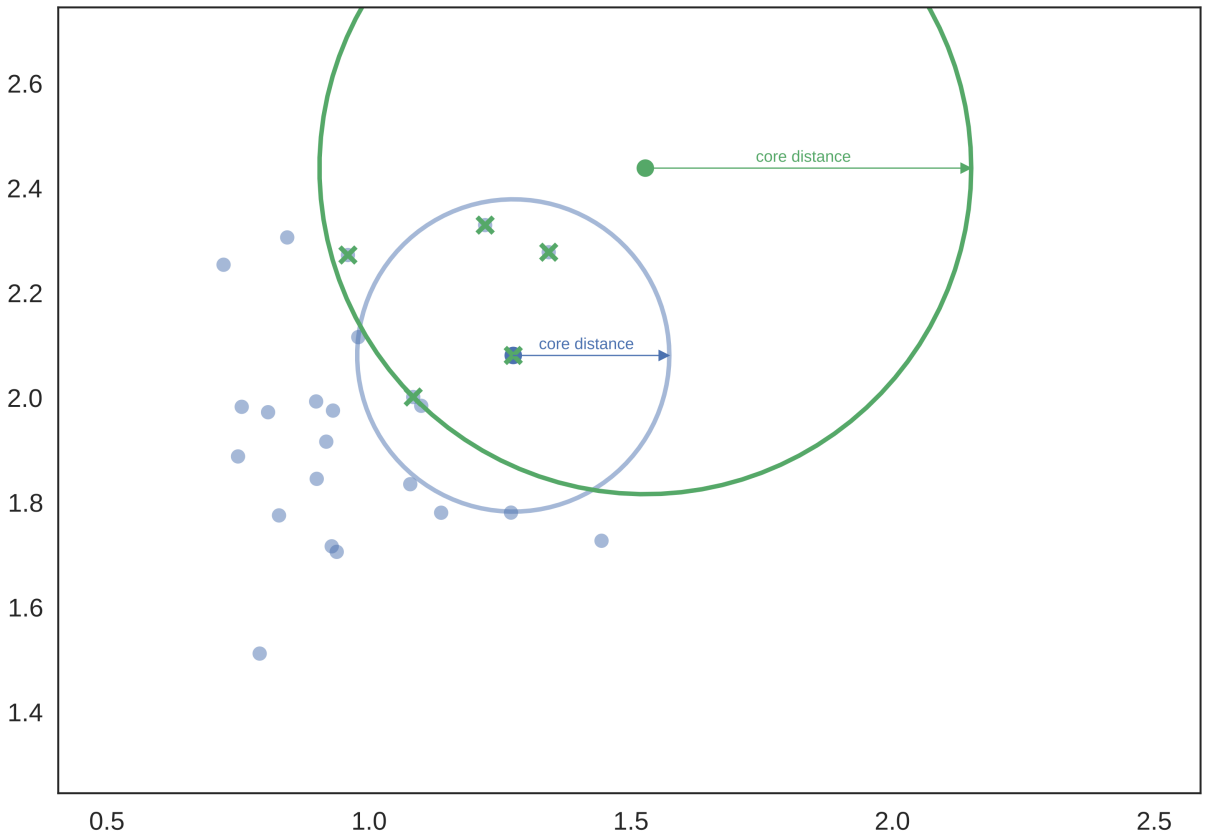
Given some $k \in \mathbb{Z}^+$, for a data point x, the $\text{core}_k(x)$ is the distance to the k-nearest neighbor of x.

Note that this definition depends on which distance we are using. Usually we will use the ℓ^2 norm.
If the language seems strange, it is inspired by the HDBSCAN algorithm itself.

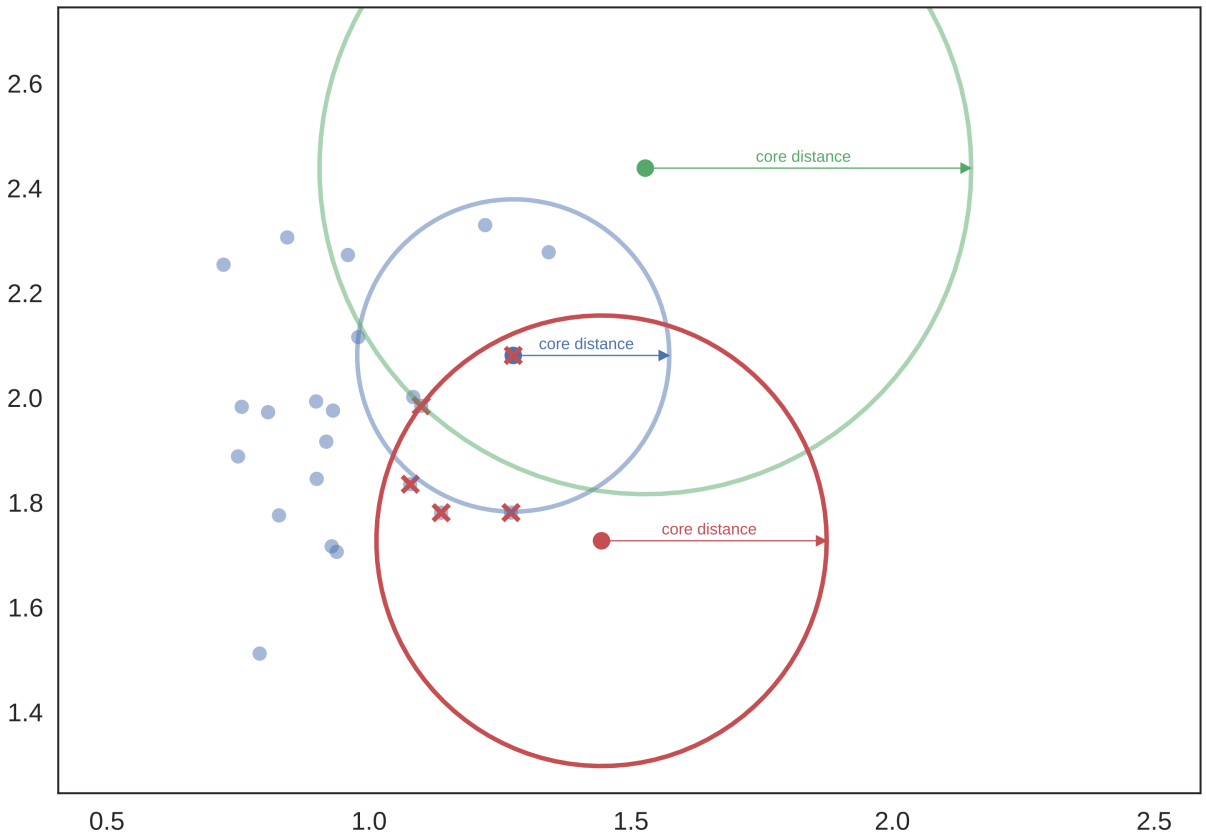
core_k Illustration



core_k Illustration



core_k Illustration



Mutual Reachability Distance

Definition

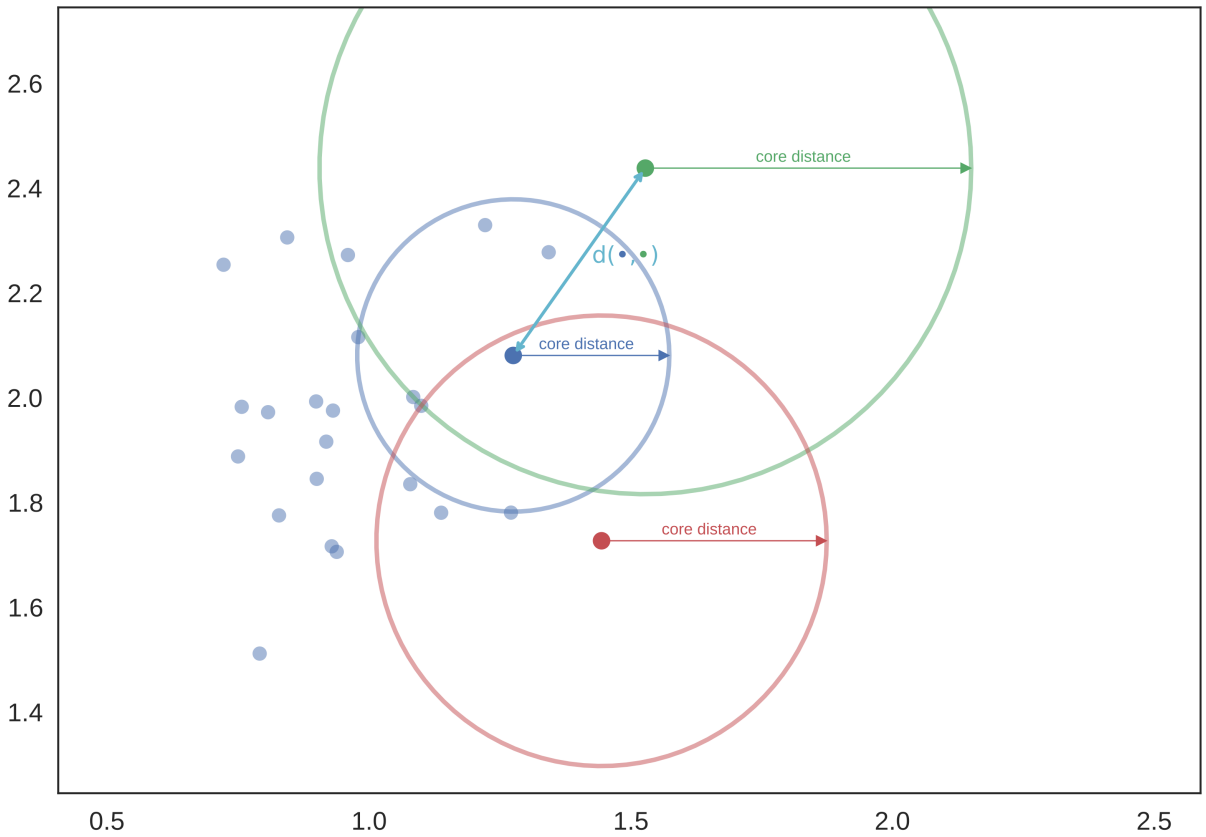
The **mutual reachability distance** of two data points x and y, written $\text{mrd}(x, y)$, with a paramater $k \in \mathbb{Z}^+$ and a metric dist, is

$$\max \begin{cases} \text{dist}(x, y): & \text{Distance between } x \text{ and } y \\ \text{core}_k(x): & \text{Distance to } k\text{-nearest neighbor of } x \\ \text{core}_k(y): & \text{Distance to } k\text{-nearest neighbor of } y \end{cases}$$

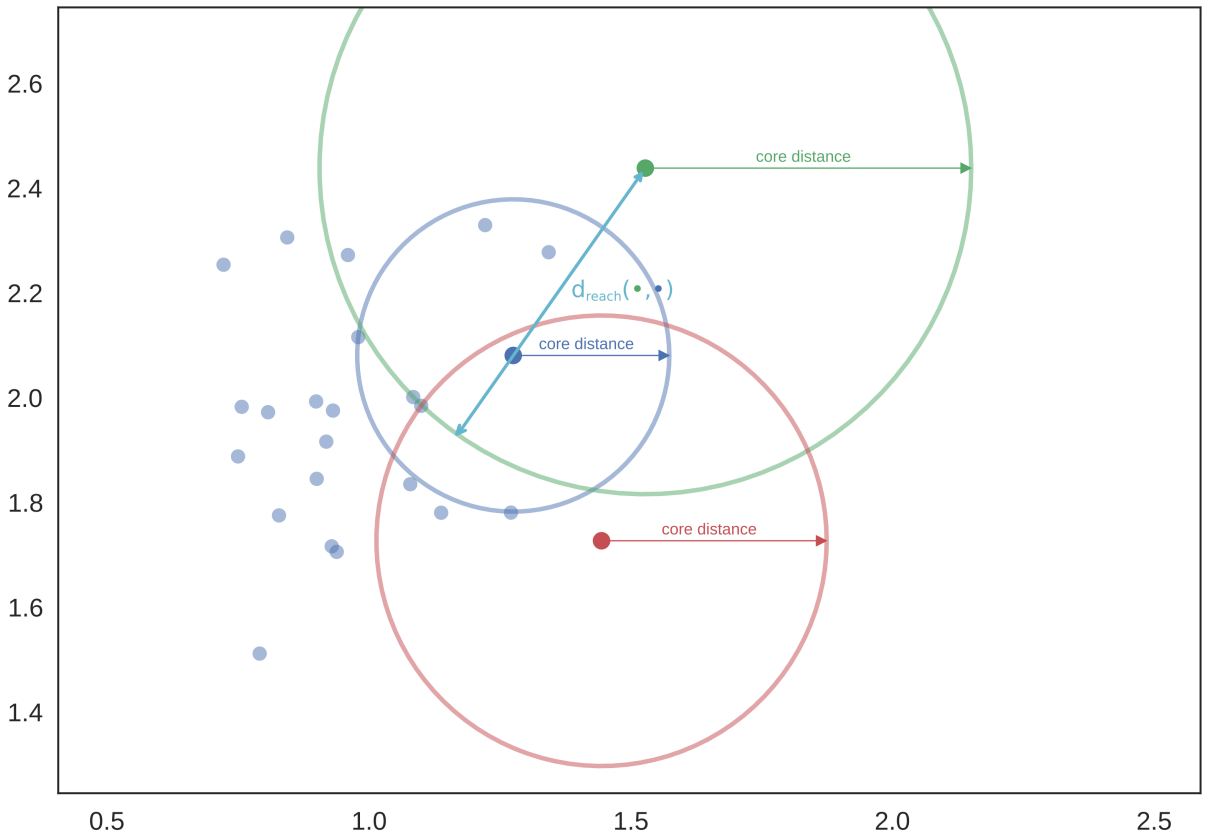
where "distance" refers to the standard ℓ^2 norm of $x - y$, or euclidian distance (For our purposes).

Note: The k-nearest neighbor is the kth closest point *not counting the point itself*. Undefined if there are k or less total data points.

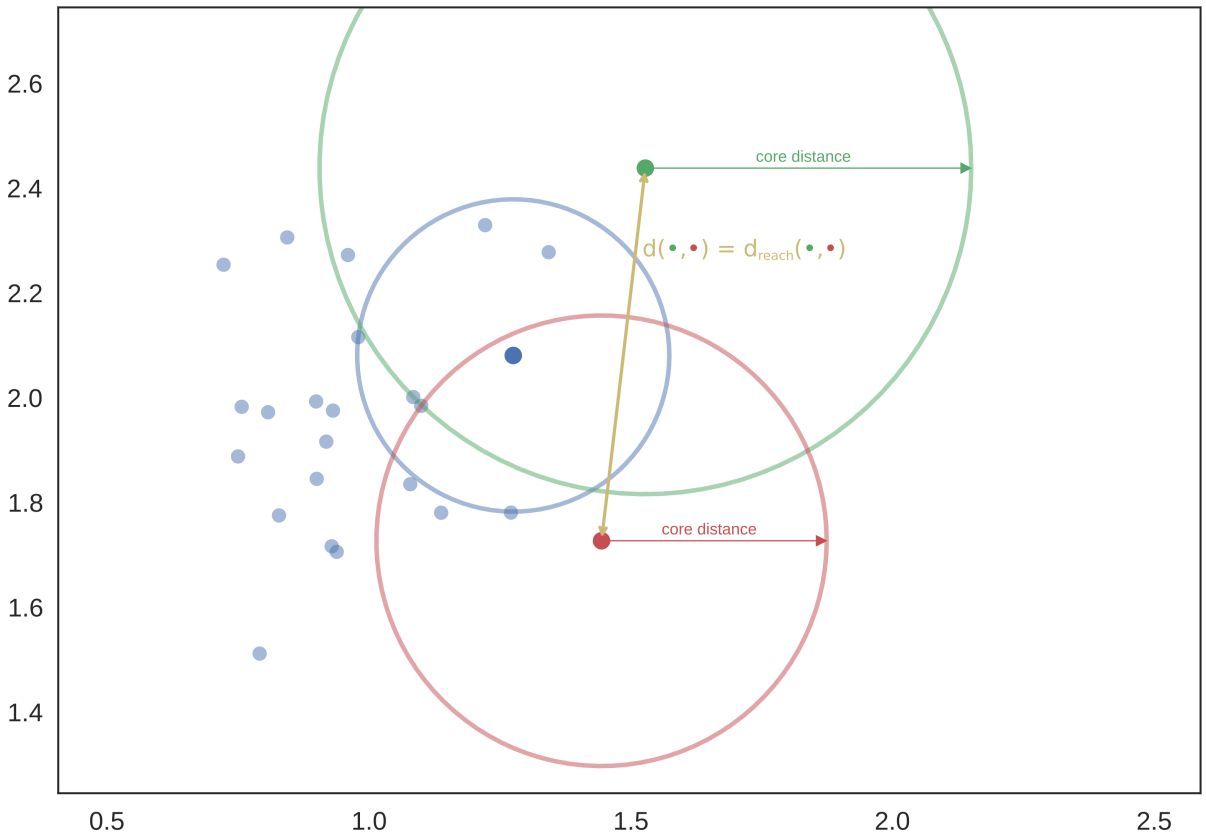
core_k Illustration



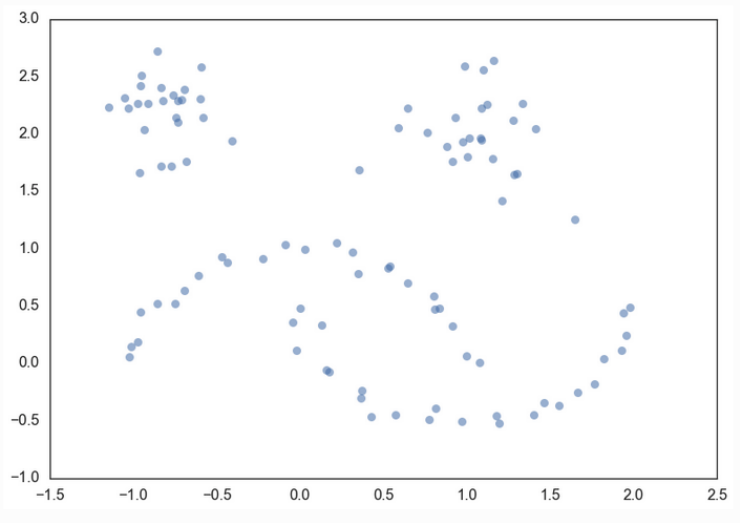
core_k Illustration



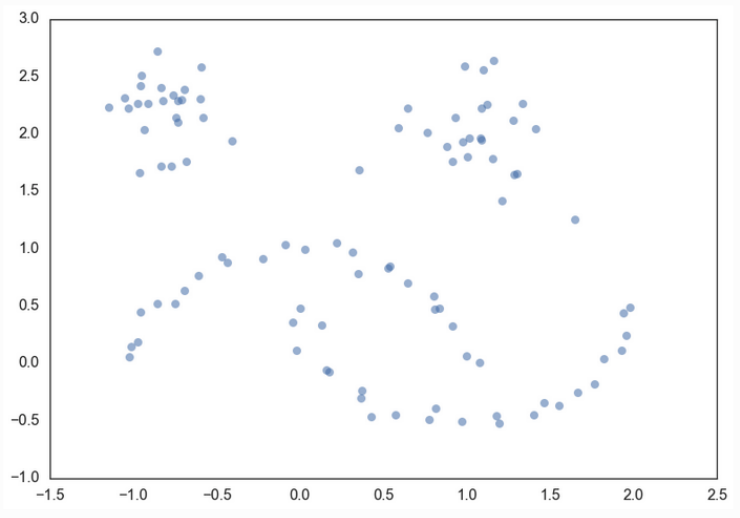
core_k Illustration

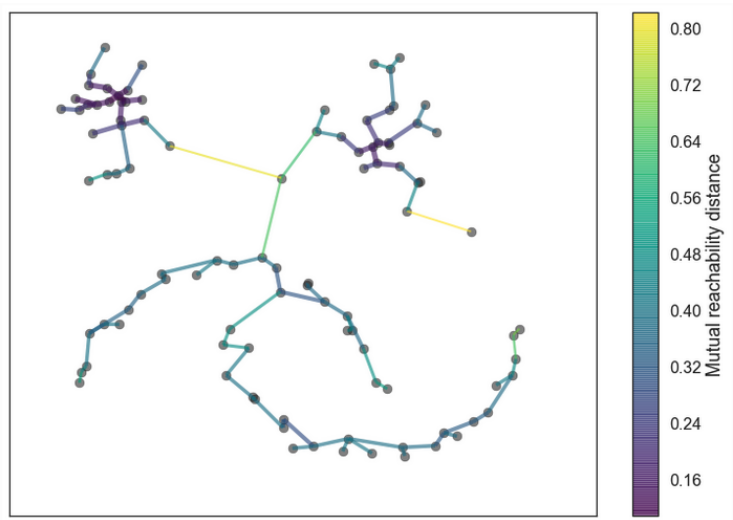


HDBSCAN

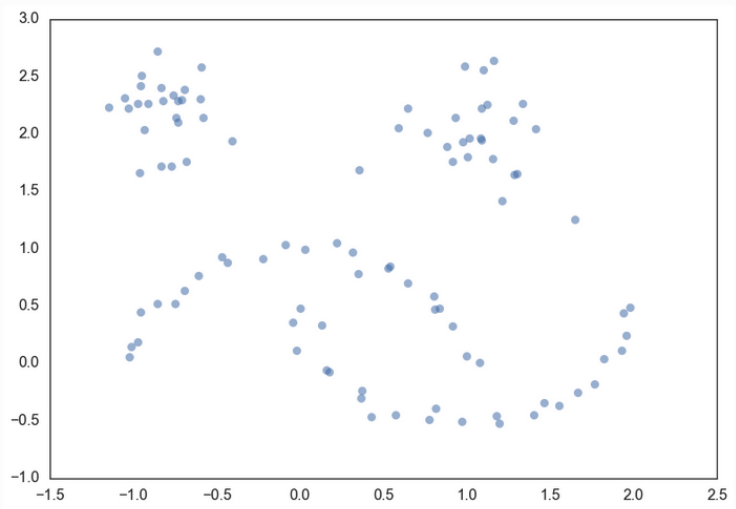


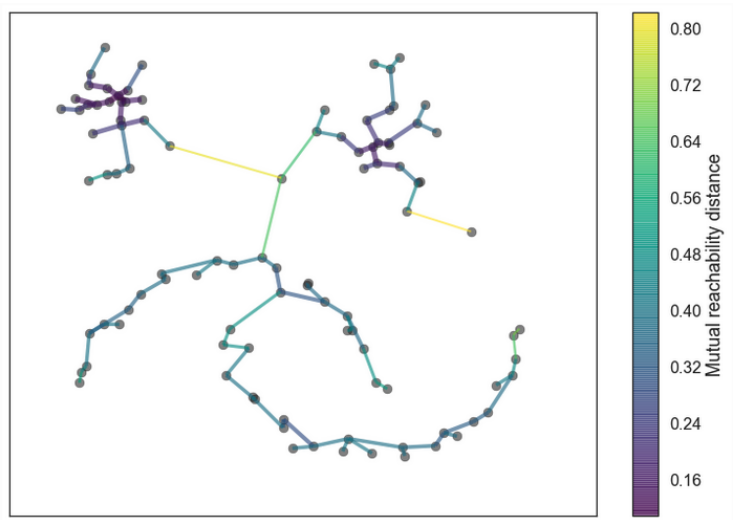
HDBSCAN

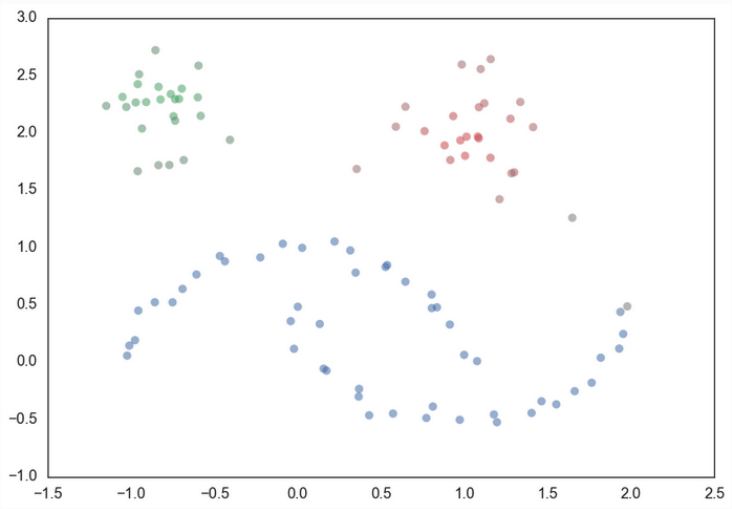

3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5


0.80
0.72
0.64
0.56
0.48
0.40
0.32
0.24
0.16
Mutual reachability distance

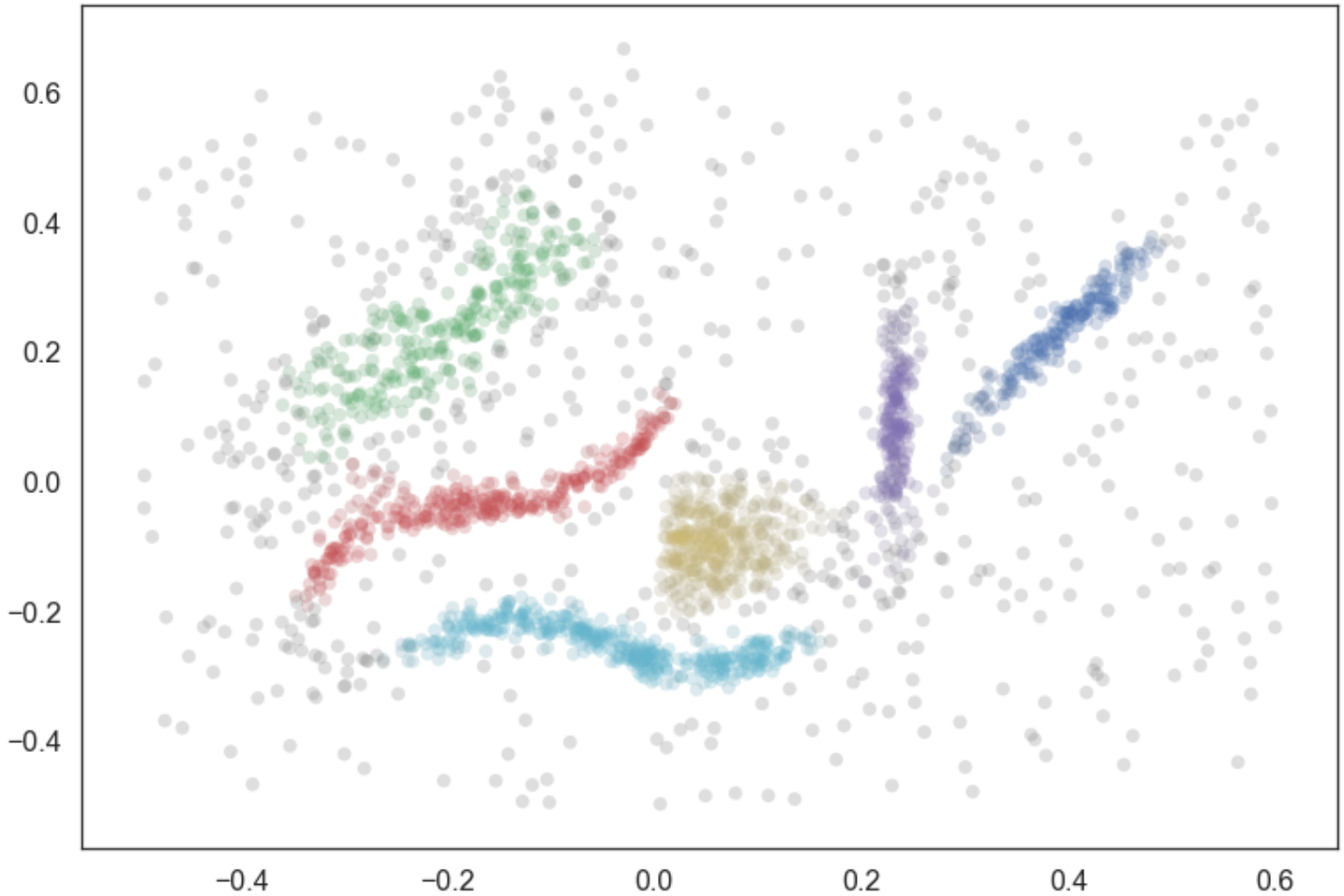
HDBSCAN


3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5

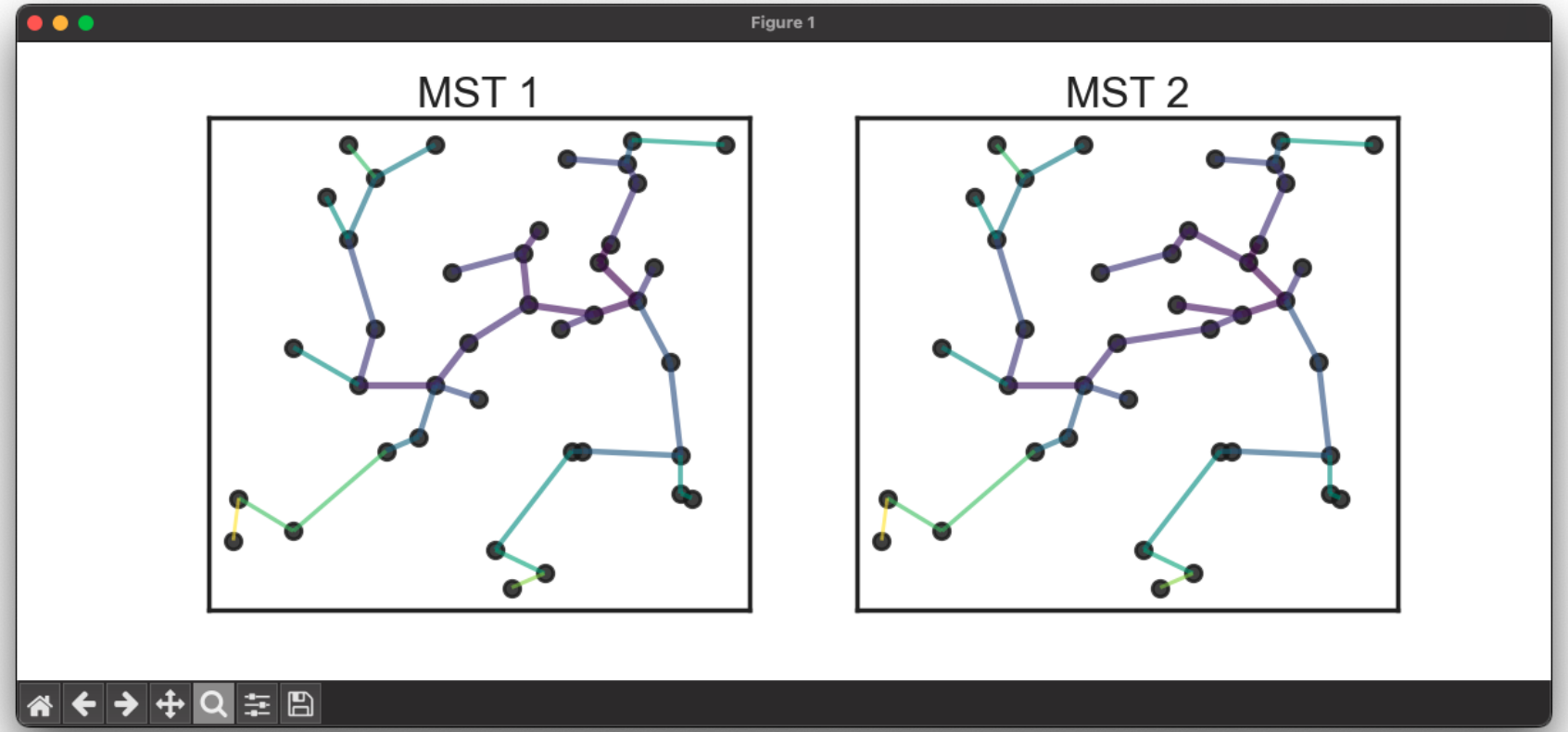

0.80
0.72
0.64
0.56
0.48
0.40
0.32
0.24
0.16
Mutual reachability distance


3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5

Example HDBSCAN Run

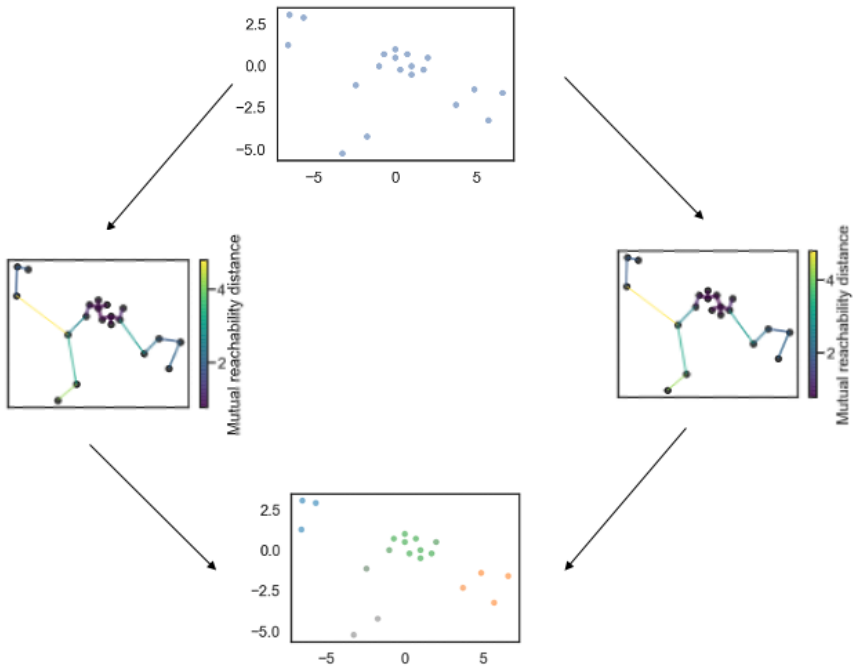


Forty Random Points



"Unreasonable Stability"

In [1], Dr. Blevins and her advisor Dr. Bridges of Oak Ridge National Labratory showed that distinct minimum spanning trees on the same data set give the same clusters.



Goals

1. Verify that it is possible to get different minimum spanning trees over the same data

Goals

1. Verify that it is possible to get different minimum spanning trees over the same data
2. Test whether this can occur in real-world data (and get a rough sense of how often)

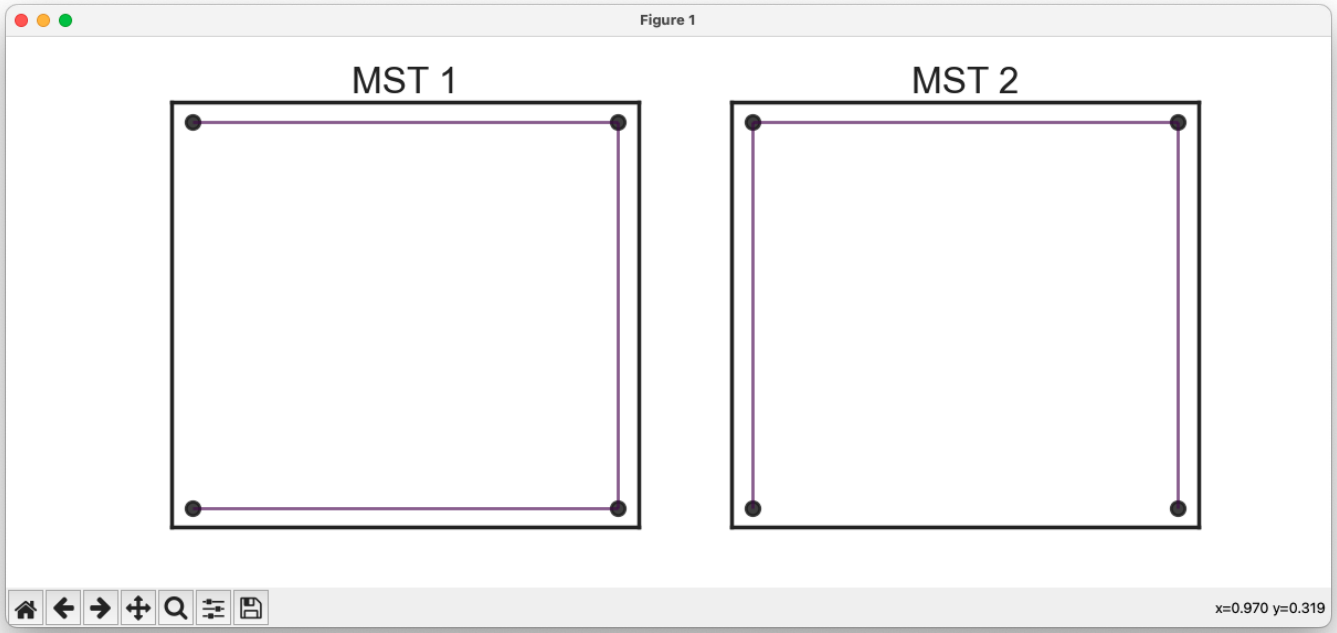
Goals

1. Verify that it is possible to get different minimum spanning trees over the same data
2. Test whether this can occur in real-world data (and get a rough sense of how often)
3. Describe sufficient conditions for non-unique minimum spanning trees

Base Case

Example

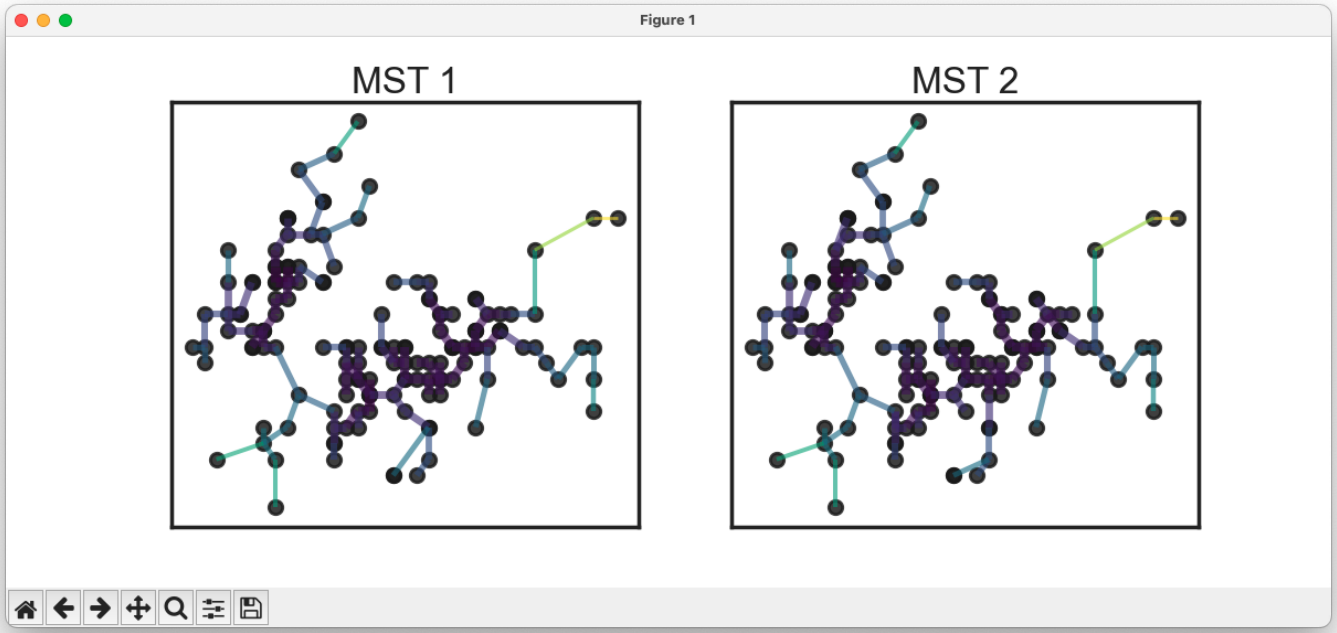
Different orderings of the simple data set $[[0, 0], [0, 1], [1, 1], [1, 0]]$ give different minimum spanning trees



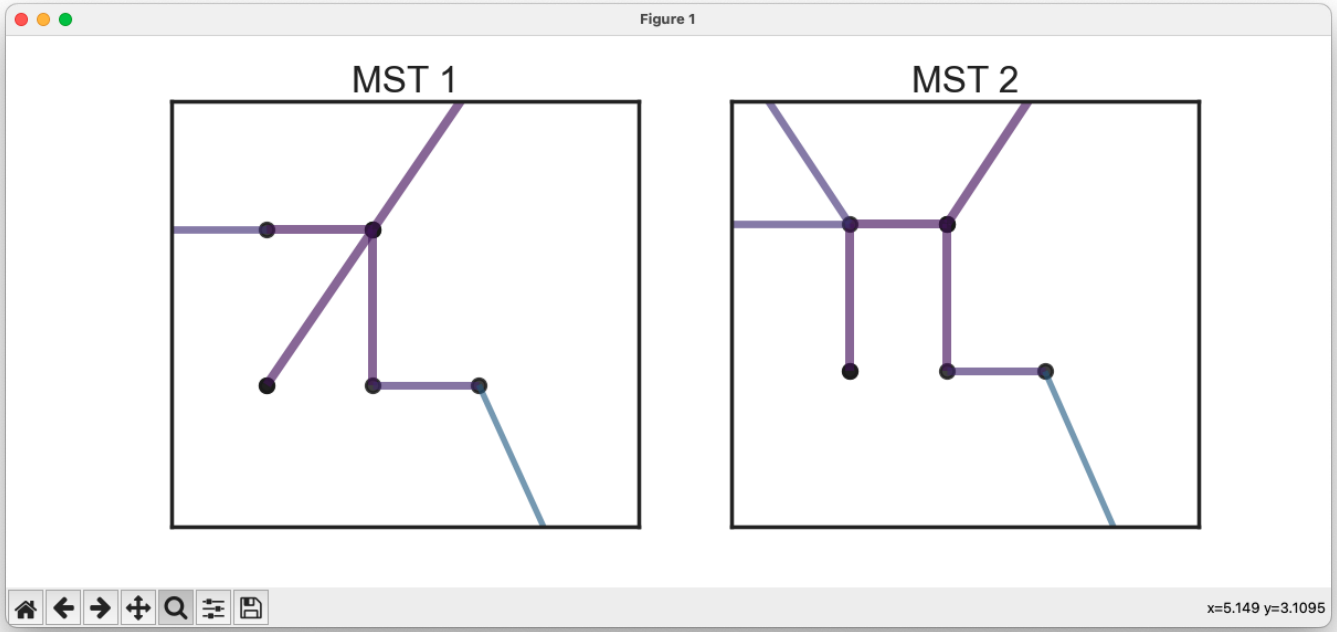
Iris

Example

the Iris data set [6] records information about various types of iris flowers.



Iris Closeup

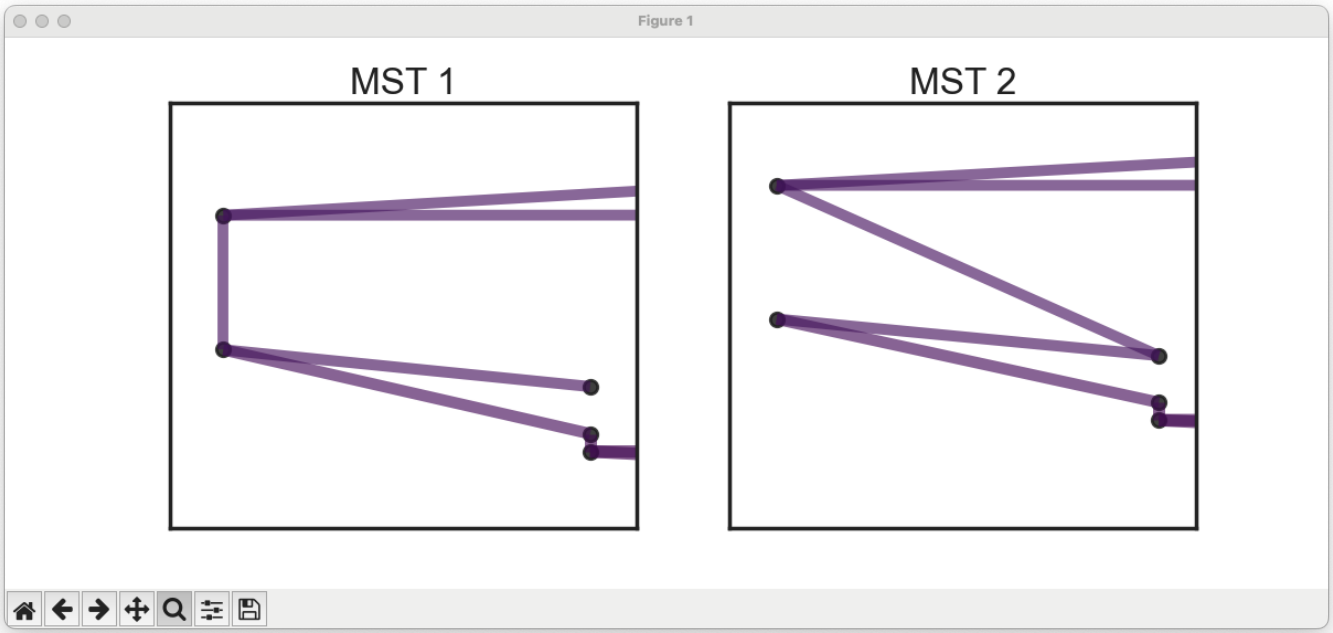


Note: All graphs (except the first) are a two-dimensional projection of the higher-dimensional data sets, but the python script verifies that there are truly multiple trees in the native dimension.

MNIST Closeup

Example

MNIST is a massive data set of handwritten digits. We plotted the first 50.



NECkSc

Definition

Let X, along with some distance definition dist, be a finite metric space. And, with a paramater $k \in \mathbb{Z}^+$, define

$$\epsilon = \min_{x \in X} \text{core}_k(x) = \text{core}_k(x_0) \text{ for some } x_0 \in X.$$

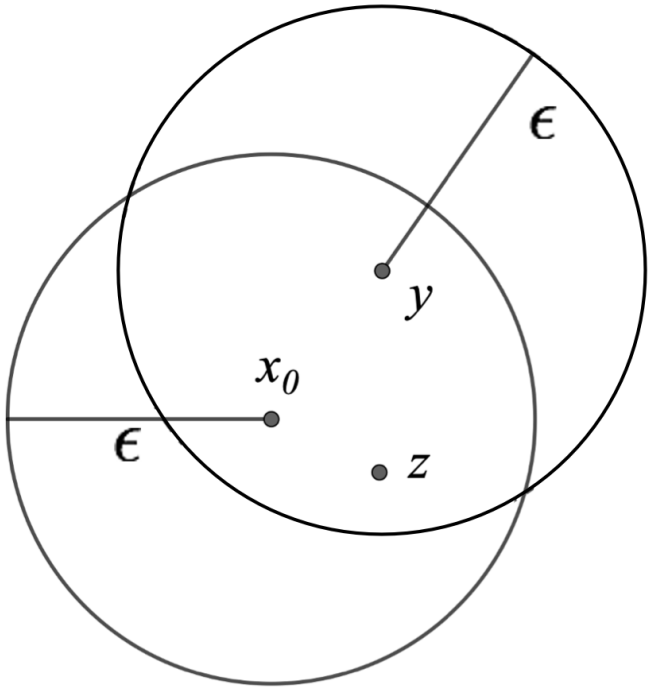
Then X is a 'not-everywhere core$_k$ sparse (closed)' (**NECkSc**) metric space, if and only if

$$\exists y, z \in \{x \in X : 0 < \text{dist}(x, x_0) \leq \epsilon\} \text{ with } y \neq z \text{ and } \text{dist}(y, z) \leq \epsilon.$$

In other words, a metric space is NECkSc if there are two points in the k-nearest neighbors of x_0 that are also within ϵ of each other.

Original NECkS Definition

Note: This is a tweaked version of a NECkS metric space.



NECkSc $\implies$ Multiple Minimum Spanning Trees

Theorem

(From [1]) Let (X, dist) be a NECkS metric space for some $k \in \mathbb{N}$. Let $G(V, E)$ be the complete graph such that $V = X$ and for $(a, b) \in E$, $w(a, b) = \text{mrd}(a, b)$. Then G has multiple minimum spanning trees.

Dr. Blevins and Dr. Bridges showed this for NECkS spaces, but it also holds for NECkSc spaces. So we need only to show that a data set (along with a definition of distance) is a NECkSc metric space to show that it has multiple minimum spanning trees.

Sufficient k value

Theorem

For $k \geq 6$, any two dimensional data set, with the ℓ^2 norm as distance, is a NECkSc space, and thus has multiple minimum spanning trees.

Proof Sketch

Increasing k cannot decrease ϵ, so proving the theorem for $k = 6$ is sufficient to show it's truth for all $k \geq 6$.

Proof Sketch

Increasing k cannot decrease ϵ, so proving the theorem for $k = 6$ is sufficient to show it's truth for all $k \geq 6$.

- Assume BWOC there exists some two-dimensional data set X that is not a NECkSc metric space for $k = 6$ (with standard euclidian distance).

Proof Sketch

Increasing k cannot decrease ϵ, so proving the theorem for $k = 6$ is sufficient to show it's truth for all $k \geq 6$.

- Assume BWOC there exists some two-dimensional data set X that is not a NECkSc metric space for $k = 6$ (with standard euclidian distance).
- Order the six closest points to x_0 in terms of their angle relative to x_0.

Proof Sketch

Increasing k cannot decrease ϵ, so proving the theorem for $k = 6$ is sufficient to show it's truth for all $k \geq 6$.

- Assume BWOC there exists some two-dimensional data set X that is not a NECkSc metric space for $k = 6$ (with standard euclidian distance).
- Order the six closest points to x_0 in terms of their angle relative to x_0.
- By the definition of $\text{core}_6(x_0) = \epsilon$, these are all within ϵ of x_0.

Proof Sketch

Increasing k cannot decrease ϵ, so proving the theorem for $k = 6$ is sufficient to show it's truth for all $k \geq 6$.

- Assume BWOC there exists some two-dimensional data set X that is not a NECkSc metric space for $k = 6$ (with standard euclidian distance).
- Order the six closest points to x_0 in terms of their angle relative to x_0.
- By the definition of $\text{core}_6(x_0) = \epsilon$, these are all within ϵ of x_0.
- No pair of these points is within ϵ of each other, becuase X is not a NECkSc space.

Proof Sketch (continued)

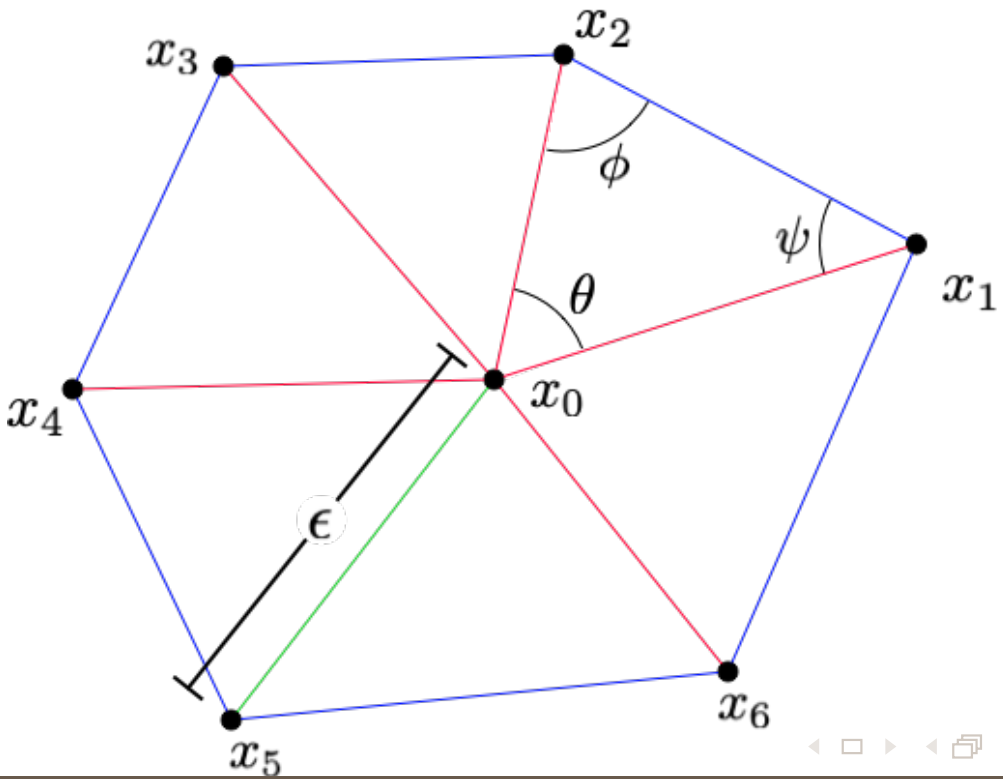
- This means that blue lines are strictly greater than ϵ, and red/green lines are less than or equal to ϵ.

Proof Sketch (continued)

- This means that blue lines are strictly greater than ϵ, and red/green lines are less than or equal to ϵ.
- The angles formed around x_0 sum to more than 2π.

Proof Sketch (continued)

- This means that blue lines are strictly greater than ϵ, and red/green lines are less than or equal to ϵ.
- The angles formed around x_0 sum to more than 2π.



The Kissing Number

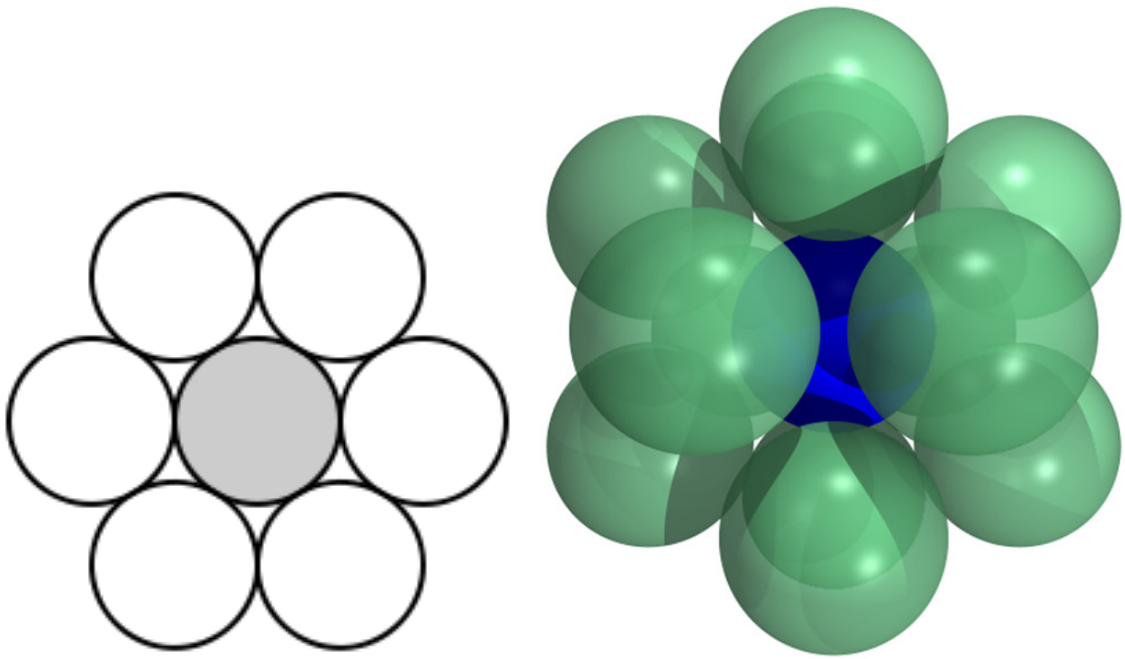
Definition

Given an n-dimensional sphere S with radius r, the n-**dimensional kissing number** kiss(n) is the maximum number of spheres of radius r that can be tangent to S without the interiors of any two spheres overlapping*.

*Two spheres' interiors overlap if their intersection is neither the empty set nor a single point.

Background on the Kissing Number

Already hard in three dimensions: Newton disagreed with David Gregory about kiss(3). "Extra" space in 3D unlike in 2D.



Kissing Numbers (from [3])

Dimension	Lower bound	Upper bound	Ratio	References
1	2		1	
2	6		1	
3	12		1	[18]
4	24		1	[17, 14]
5	40	44	1.100	[7, 13]
6	72	78	1.084	[7, 13]
7	126	134	1.064	[7, 13]
8	240		1	[7, 11, 16]
9	306	363	1.187	[10, 12]
10	510	553	1.085	[5, 12]
11	592	868	1.467	[5, 8]
12	840	1355	1.614	[10, 8]

$k > \text{kiss}(n) \implies \text{NEC}k\text{Sc}$

Theorem

Let X be n-dimensional data set, and let dist *be euclidian distance. Then (X, dist) is a NECkSc space if*

1. $k > \text{kiss}(n)$ *and*
2. $|X| > k^*$

*This second point is formally necessary but practically meaningless: $|X| \leq k$ would mean the data is all in one cluster.

Proof Sketch

Let X be an n-dimensional data set, and let (X, dist) **not** be a NECkSc space. We show that this implies $k \leq \text{kiss}(n)$.

Proof Sketch

Let X be an n-dimensional data set, and let (X, dist) **not** be a NECkSc space. We show that this implies $k \leq \text{kiss}(n)$.

- Consider the k closest points to x_0, call this set A.

Proof Sketch

Let X be an n-dimensional data set, and let (X, dist) **not** be a NECkSc space. We show that this implies $k \leq \text{kiss}(n)$.

- Consider the k closest points to x_0, call this set A.
- Notice that $|A| = k$.

Proof Sketch

Let X be an n-dimensional data set, and let (X, dist) **not** be a NECkSc space. We show that this implies $k \leq \text{kiss}(n)$.

- Consider the k closest points to x_0, call this set A.
- Notice that $|A| = k$.
- Let each $p \in A$ correspond to a new p', on $\overrightarrow{x_0 p}$, with $\text{dist}(x_0, p') = \epsilon \geq \text{dist}(x_0, p)$. (Call this set of new points A').

Proof Sketch

Let X be an n-dimensional data set, and let (X, dist) **not** be a NECkSc space. We show that this implies $k \leq \text{kiss}(n)$.

- Consider the k closest points to x_0, call this set A.
- Notice that $|A| = k$.
- Let each $p \in A$ correspond to a new p', on $\overrightarrow{x_0 p}$, with $\text{dist}(x_0, p') = \epsilon \geq \text{dist}(x_0, p)$. (Call this set of new points A').
- $|A'| = |A| = k$.

Proof Sketch

Let X be an n-dimensional data set, and let (X, dist) **not** be a NECkSc space. We show that this implies $k \leq \text{kiss}(n)$.

- Consider the k closest points to x_0, call this set A.
- Notice that $|A| = k$.
- Let each $p \in A$ correspond to a new p', on $\overrightarrow{x_0 p}$, with $\text{dist}(x_0, p') = \epsilon \geq \text{dist}(x_0, p)$. (Call this set of new points A').
- $|A'| = |A| = k$.
- For all $p, q \in A$ with $p \neq q$, $\text{dist}(p, q) > \epsilon$, becuase (X, dist) is not a NECkSc space.

Proof Sketch

Let X be an n-dimensional data set, and let (X, dist) **not** be a NECkSc space. We show that this implies $k \leq \text{kiss}(n)$.

- Consider the k closest points to x_0, call this set A.
- Notice that $|A| = k$.
- Let each $p \in A$ correspond to a new p', on $\overrightarrow{x_0 p}$, with $\text{dist}(x_0, p') = \epsilon \geq \text{dist}(x_0, p)$. (Call this set of new points A').
- $|A'| = |A| = k$.
- For all $p, q \in A$ with $p \neq q$, $\text{dist}(p, q) > \epsilon$, becuase (X, dist) is not a NECkSc space.
- Because new points travel outward along diverging lines, we also know $\text{dist}(p', q') > \epsilon$ for all $p', q' \in A'$ with $p' \neq q'$.

Proof Sketch (continued)

- We place a hypersphere of radius $\frac{\epsilon}{2}$ centered at each $p' \in A'$, as well as one centered at x_0.

Proof Sketch (continued)

- We place a hypersphere of radius $\frac{\epsilon}{2}$ centered at each $p' \in A'$, as well as one centered at x_0.
- These hyperspheres are all tangent to the hypersphere centered at x_0, because for each $p' \in A'$, $\text{dist}(p', x_0) = \epsilon$.

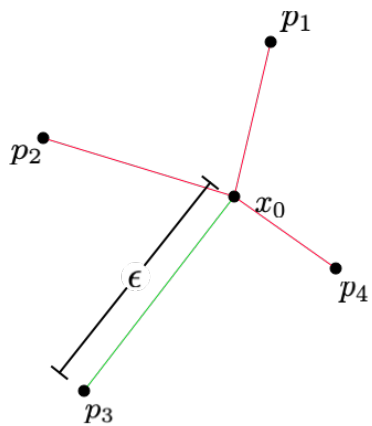
Proof Sketch (continued)

- We place a hypersphere of radius $\frac{\epsilon}{2}$ centered at each $p' \in A'$, as well as one centered at x_0.
- These hyperspheres are all tangent to the hypersphere centered at x_0, because for each $p' \in A'$, $\text{dist}(p', x_0) = \epsilon$.
- But no hypersphere is tangent to any other, because $\text{dist}(p', q') > \epsilon$ for all $p', q' \in A'$ with $p' \neq q'$.

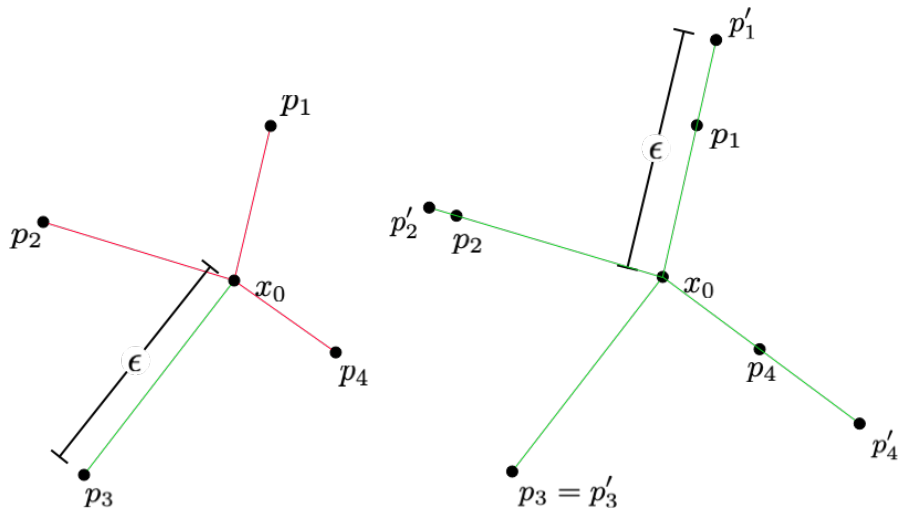
Proof Sketch (continued)

- We place a hypersphere of radius $\frac{\epsilon}{2}$ centered at each $p' \in A'$, as well as one centered at x_0.
- These hyperspheres are all tangent to the hypersphere centered at x_0, because for each $p' \in A'$, $\text{dist}(p', x_0) = \epsilon$.
- But no hypersphere is tangent to any other, because $\text{dist}(p', q') > \epsilon$ for all $p', q' \in A'$ with $p' \neq q'$.
- These all have radius $\frac{\epsilon}{2}$, so $\text{kiss}(n) \geq k$ by definition.

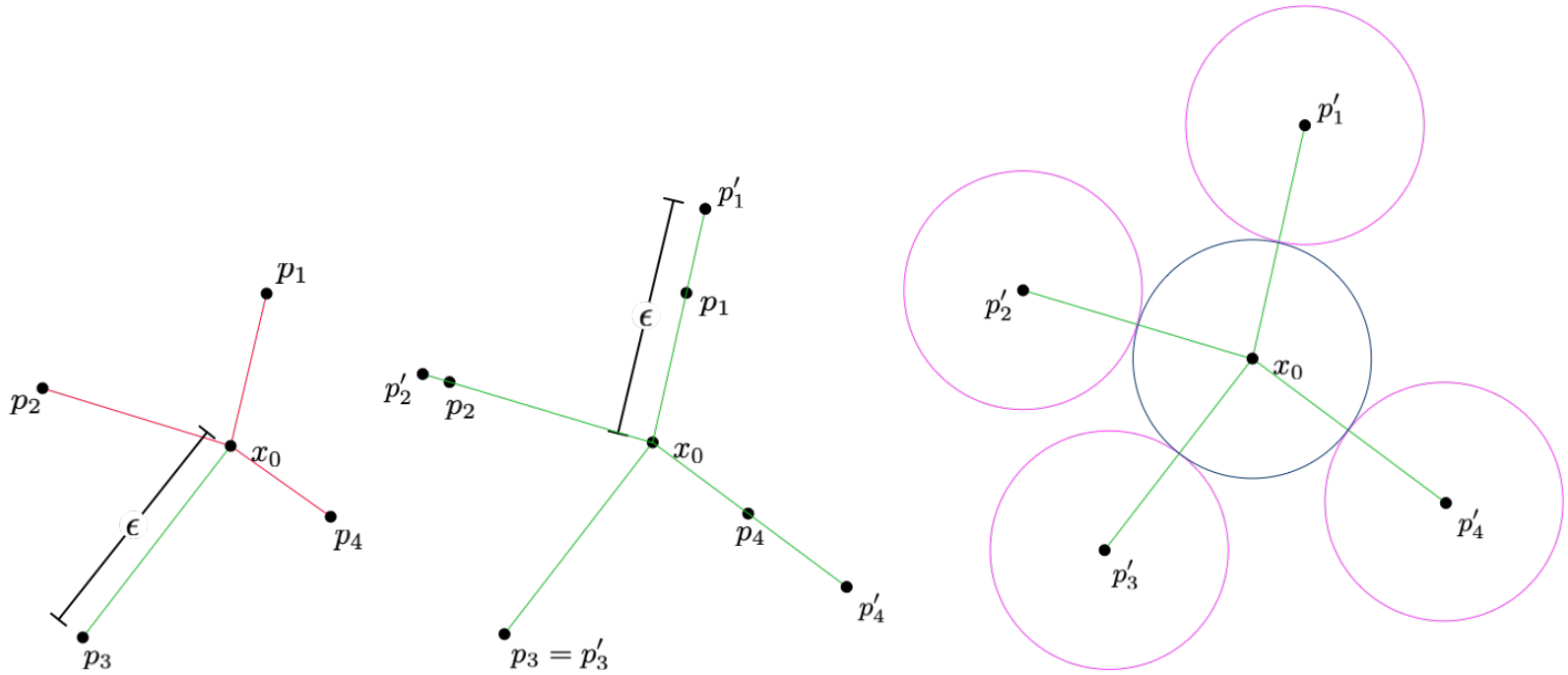
Proof Illustration in Two Dimensions



Proof Illustration in Two Dimensions



Proof Illustration in Two Dimensions



Conclusion

- We have verified that it is possible to get different minimum spanning trees from HDBSCAN over the same data set.

Conclusion

- We have verified that it is possible to get different minimum spanning trees from HDBSCAN over the same data set.
- We have tested that this can occur in real-world data, and have empirically shown that this is often the case.

Conclusion

- We have verified that it is possible to get different minimum spanning trees from HDBSCAN over the same data set.
- We have tested that this can occur in real-world data, and have empirically shown that this is often the case.
- We have shown conditions that gurantee a metric space is NECkSc, which imply non-unique minimum spanning trees.

References

Deborah Blevins.

The unreasonable stability of hdbscan.

Presentation, 2017.

Ricardo Campello, Davoud Moulavi, and Arthur Zimek.

Hierarchical density estimates for data clustering, visualization, and outlier detection.

ACM Transactions on Knowledge Discovery from Data, 10(5), 2015.

Henry Cohn.

Kissing numbers.

https://cohn.mit.edu/kissing-numbers, 2023.

Dariel Dato-On.

Mnist in csv.

https://www.kaggle.com/datasets/oddrationale/mnist-in-csv, 2018.

Leland McInnes, John Healy, and Steve Astels.

How hdbscan works.

https://hdbscan.readthedocs.io/en/latest/how_hdbscan_works.html, 2016.

Antony Unwin and Kim Kleinman.

The iris data set: In search of the source of virginica.

Significance, 18, 2021.

Sports Come Up in a Data Clustering Presentation (Again)