Distance in High Dimensions and Clustering

Chris Cornwell

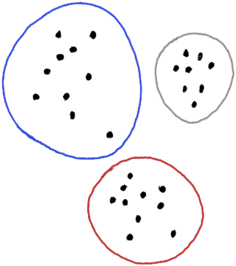
April 29, 2025

Outline

Outline

Clustering uses Distance

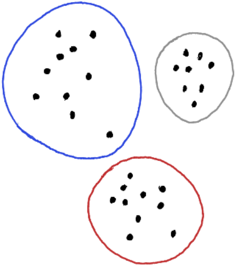
Aim of clustering: group the training data $\mathcal{S}$ into **clusters** $C_1, C_2, \ldots, C_k$, with every point in some cluster C_i (i.e., $\mathcal{S} = C_1 \cup C_2 \cup \ldots \cup C_k$) and clusters are disjoint.[a]



Clustering uses Distance

Aim of clustering: group the training data $\mathcal{S}$ into **clusters** $C_1, C_2, \ldots, C_k$, with every point in some cluster C_i (i.e., $\mathcal{S} = C_1 \cup C_2 \cup \ldots \cup C_k$) and clusters are disjoint.[a]

Goal: have points in the same cluster to be "similar".

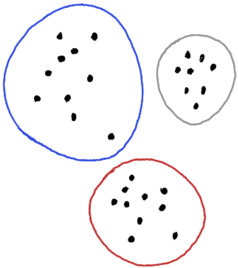


Clustering uses Distance

Aim of clustering: group the training data $\mathcal{S}$ into **clusters** $C_1, C_2, \ldots, C_k$, with every point in some cluster C_i (i.e., $\mathcal{S} = C_1 \cup C_2 \cup \ldots \cup C_k$) and clusters are disjoint.[a]

Goal: have points in the same cluster to be "similar".

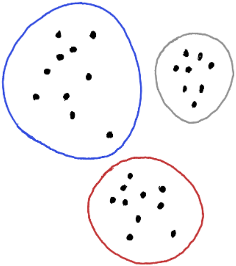
While most types of ML algorithms are affected by how densely points are packed in $\mathcal{S}$, clustering algorithms typically use distance (to the nearest points in $\mathcal{S}$) to measure similarity of points.



[a] These conditions are the common ones. However, sometimes one may wish to exclude an *outlier* from being in a cluster, and sometimes clusters are "fuzzy," meaning that points have a probability for being in each cluster.

Clustering uses Distance

Aim of clustering: group the training data $\mathcal{S}$ into **clusters** $C_1, C_2, \ldots, C_k$, with every point in some cluster C_i (i.e., $\mathcal{S} = C_1 \cup C_2 \cup \ldots \cup C_k$) and clusters are disjoint.[a]



Goal: have points in the same cluster to be "similar".

While most types of ML algorithms are affected by how densely points are packed in $\mathcal{S}$, clustering algorithms typically use distance (to the nearest points in $\mathcal{S}$) to measure similarity of points.

[a] These conditions are the common ones. However, sometimes one may wish to exclude an *outlier* from being in a cluster, and sometimes clusters are "fuzzy," meaning that points have a probability for being in each cluster.

- Makes a phenomenon called the **curse of dimensionality** especially relevant.

What is the Curse of Dimensionality?

Not universally, clearly defined what falls under the umbrella of the curse of dimensionality and what does not.

What is the Curse of Dimensionality?

Not universally, clearly defined what falls under the umbrella of the curse of dimensionality and what does not.

- Strict interpretation: The amount of training data used needs to increase exponentially in the number of features, i.e., independent variables. (If the number of samples needed to see how position/value of one feature might affect y labeling is roughly constant over the features.)

What is the Curse of Dimensionality?

Not universally, clearly defined what falls under the umbrella of the curse of dimensionality and what does not.

- Strict interpretation: The amount of training data used needs to increase exponentially in the number of features, i.e., independent variables. (If the number of samples needed to see how position/value of one feature might affect y labeling is roughly constant over the features.)
- Broad interpretation: With large number of features (so, large d, where $\mathbf{x}_i \in \mathbb{R}^d$), our intuition for the way that the distance between points relates to properties we care about will break down.
  Distance in high dimensions is *weird*. (*Let's see how*.)

Spheres in $\mathbb{R}^d$, d large: weird

Often, work with those points that are within a given distance R from fixed point. These are points in a d-dimensional "ball" (that is, enclosed by a d-dimensional sphere):

$$B_R(\mathbf{p}) = \{\mathbf{x} \in \mathbb{R}^d \mid |\mathbf{x} - \mathbf{p}| \leq R\}.$$

Spheres in $\mathbb{R}^d$, d large: weird

Often, work with those points that are within a given distance R from fixed point. These are points in a d-dimensional "ball" (that is, enclosed by a d-dimensional sphere):

$$B_R(\mathbf{p}) = \{\mathbf{x} \in \mathbb{R}^d \mid |\mathbf{x} - \mathbf{p}| \leq R\}.$$

(*Distance* here is $|\mathbf{x} - \mathbf{p}| = \sqrt{\sum_{i=1}^{d} (x_i - p_i)^2}$, usual Euclidean norm.)

Spheres in $\mathbb{R}^d$, d large: weird

Often, work with those points that are within a given distance R from fixed point. These are points in a d-dimensional "ball" (that is, enclosed by a d-dimensional sphere):

$$B_R(\mathbf{p}) = \{\mathbf{x} \in \mathbb{R}^d \mid |\mathbf{x} - \mathbf{p}| \leq R\}.$$

(*Distance* here is $|\mathbf{x} - \mathbf{p}| = \sqrt{\sum_{i=1}^{d}(x_i - p_i)^2}$, usual Euclidean norm.)

The volume of $B_R(\mathbf{p})$: $\frac{\pi^{d/2}}{\Gamma(\frac{d}{2}+1)} R^d$.

Γ is Euler's gamma function. If d is even, $\Gamma(\frac{d}{2} + 1) = (\frac{d}{2})!$ and if d is odd, it's roughly similar: $(\frac{d}{2})(\frac{d}{2} - 1) \ldots (\frac{1}{2})\pi^{1/2}$.

Spheres in $\mathbb{R}^d$, d large: weird

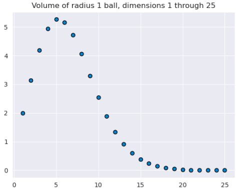
Often, work with those points that are within a given distance R from fixed point. These are points in a d-dimensional "ball" (that is, enclosed by a d-dimensional sphere):

$$B_R(\mathbf{p}) = \{\mathbf{x} \in \mathbb{R}^d \mid |\mathbf{x} - \mathbf{p}| \le R\}.$$

(*Distance* here is $|\mathbf{x} - \mathbf{p}| = \sqrt{\sum_{i=1}^d (x_i - p_i)^2}$, usual Euclidean norm.)

The volume of $B_R(\mathbf{p})$: $\frac{\pi^{d/2}}{\Gamma(\frac{d}{2}+1)} R^d$.

Γ is Euler's gamma function. If d is even, $\Gamma(\frac{d}{2}+1) = (\frac{d}{2})!$ and if d is odd, it's roughly similar: $(\frac{d}{2})(\frac{d}{2}-1)\ldots(\frac{1}{2})\pi^{1/2}$.



Most points are near boundary

So, for any fixed radius $R > 0$, the volume of the d-dimensional $B_R(\mathbf{p})$ approaches 0 as $d \longrightarrow \infty$.

Most points are near boundary

So, for any fixed radius $R > 0$, the volume of the d-dimensional $B_R(\mathbf{p})$ approaches 0 as $d \longrightarrow \infty$.
Additionally, *where* the space/volume within $B_R(\mathbf{p})$ is distributed changes as dimension increases.

Most points are near boundary

So, for any fixed radius $R > 0$, the volume of the d-dimensional $B_R(\mathbf{p})$ approaches 0 as $d \longrightarrow \infty$.

Additionally, *where* the space/volume within $B_R(\mathbf{p})$ is distributed changes as dimension increases.

Choose ε with $0 < \varepsilon < R$. What proportion of points in $B_R(\mathbf{p})$ are at least ε away from the boundary sphere? That is, how large is $B_{R-\varepsilon}(\mathbf{p})$ in comparison to $B_R(\mathbf{p})$?

Most points are near boundary

So, for any fixed radius $R > 0$, the volume of the d-dimensional $B_R(\mathbf{p})$ approaches 0 as $d \longrightarrow \infty$.

Additionally, *where* the space/volume within $B_R(\mathbf{p})$ is distributed changes as dimension increases.

Choose ε with $0 < \varepsilon < R$. What proportion of points in $B_R(\mathbf{p})$ are at least ε away from the boundary sphere? That is, how large is $B_{R-\varepsilon}(\mathbf{p})$ in comparison to $B_R(\mathbf{p})$? In dimension $d = 2$, with $R = 1$:

$$\frac{\mathrm{Vol}\, B_{1-\varepsilon}}{\mathrm{Vol}\, B_1} = \frac{\pi(1-\varepsilon)^2}{\pi} = (1-\varepsilon)^2.$$

Most points are near boundary

So, for any fixed radius $R > 0$, the volume of the d-dimensional $B_R(\mathbf{p})$ approaches 0 as $d \longrightarrow \infty$.

Additionally, *where* the space/volume within $B_R(\mathbf{p})$ is distributed changes as dimension increases.

Choose ε with $0 < \varepsilon < R$. What proportion of points in $B_R(\mathbf{p})$ are at least ε away from the boundary sphere? That is, how large is $B_{R-\varepsilon}(\mathbf{p})$ in comparison to $B_R(\mathbf{p})$? In dimension $d = 2$, with $R = 1$:

$$\frac{\operatorname{Vol} B_{1-\varepsilon}}{\operatorname{Vol} B_1} = \frac{\pi(1-\varepsilon)^2}{\pi} = (1-\varepsilon)^2.$$

For example, if $\varepsilon = 0.05$ then this is a little more than 0.9.

Most points are near boundary

So, for any fixed radius $R > 0$, the volume of the d-dimensional $B_R(\mathbf{p})$ approaches 0 as $d \longrightarrow \infty$.

Additionally, *where* the space/volume within $B_R(\mathbf{p})$ is distributed changes as dimension increases.

Choose ε with $0 < \varepsilon < R$. What proportion of points in $B_R(\mathbf{p})$ are at least ε away from the boundary sphere? That is, how large is $B_{R-\varepsilon}(\mathbf{p})$ in comparison to $B_R(\mathbf{p})$? In dimension $d = 2$, with $R = 1$:

$$\frac{\operatorname{Vol} B_{1-\varepsilon}}{\operatorname{Vol} B_1} = \frac{\pi(1-\varepsilon)^2}{\pi} = (1-\varepsilon)^2.$$

For example, if $\varepsilon = 0.05$ then this is a little more than 0.9.

Generally,

$$\frac{\operatorname{Vol} B_{R-\varepsilon}}{\operatorname{Vol} B_R} = \frac{\pi^{d/2}(R-\varepsilon)^d}{\Gamma(d/2+1)} \frac{\Gamma(d/2+1)}{\pi^{d/2}R^d} = \left(\frac{R-\varepsilon}{R}\right)^d = \left(1 - \frac{\varepsilon}{R}\right)^d.$$

Since $1 - \frac{\varepsilon}{R} < 1$, this approaches 0 as $d \longrightarrow \infty$.

Most points are near boundary

So, for any fixed radius $R > 0$, the volume of the d-dimensional $B_R(\mathbf{p})$ approaches 0 as $d \longrightarrow \infty$.

Additionally, *where* the space/volume within $B_R(\mathbf{p})$ is distributed changes as dimension increases.

Choose ε with $0 < \varepsilon < R$. What proportion of points in $B_R(\mathbf{p})$ are at least ε away from the boundary sphere? That is, how large is $B_{R-\varepsilon}(\mathbf{p})$ in comparison to $B_R(\mathbf{p})$? In dimension $d = 2$, with $R = 1$:

$$\frac{\mathrm{Vol}\, B_{1-\varepsilon}}{\mathrm{Vol}\, B_1} = \frac{\pi(1-\varepsilon)^2}{\pi} = (1-\varepsilon)^2.$$

For example, if $\varepsilon = 0.05$ then this is a little more than 0.9.

Generally,

$$\frac{\mathrm{Vol}\, B_{R-\varepsilon}}{\mathrm{Vol}\, B_R} = \frac{\pi^{d/2}(R-\varepsilon)^d}{\Gamma(d/2+1)} \frac{\Gamma(d/2+1)}{\pi^{d/2}R^d} = \left(\frac{R-\varepsilon}{R}\right)^d = \left(1 - \frac{\varepsilon}{R}\right)^d.$$

Since $1 - \frac{\varepsilon}{R} < 1$, this approaches 0 as $d \longrightarrow \infty$. Returning to $\varepsilon = 0.05$ and $R = 1$, the ratio is less than 0.05 if $d \geq 59$; so, more than 95% of the volume of $B_R(\mathbf{p})$ is contained in an outer shell, within 0.05 of the boundary.

Most points are near boundary

A consequence?

- Given a point, say you randomly select some points in data from those that are within a given distance of it.

Most points are near boundary

A consequence?

- Given a point, say you randomly select some points in data from those that are within a given distance of it.
  The higher the dimension, the more likely these points will be nearly equal distance from the chosen point. So, which *one* is closest is more subject to random chance (from a small amount of noise, say).

Most points are near boundary

A consequence?

- Given a point, say you randomly select some points in data from those that are within a given distance of it.
  The higher the dimension, the more likely these points will be nearly equal distance from the chosen point. So, which *one* is closest is more subject to random chance (from a small amount of noise, say).

Increasingly likely, also, that none of these points are near each other:

Most points are near boundary

A consequence?

- Given a point, say you randomly select some points in data from those that are within a given distance of it.
  The higher the dimension, the more likely these points will be nearly equal distance from the chosen point. So, which *one* is closest is more subject to random chance (from a small amount of noise, say).

Increasingly likely, also, that none of these points are near each other: points below sampled with random coordinates (i.i.d., with a mean-zero normal distribution). Distances between all pairs of sampled points were calculated and plotted in a histogram.

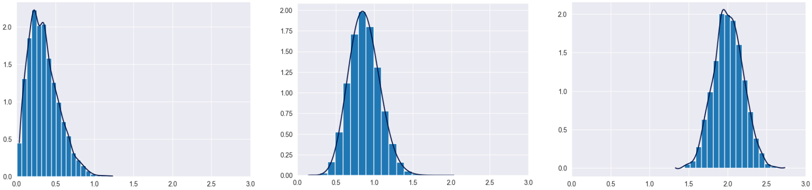


Figure: Left: points in $\mathbb{R}^2$; Middle: points in $\mathbb{R}^{10}$; Right: points in $\mathbb{R}^{50}$

Another example of high dimensional weirdness

In $\mathbb{R}^2$, consider the five depicted circles in the square $[-2,2]^2$. The four "corner" circles are tangent to (two) edges of the square and tangent to each other. Each of them has radius 1. The "center" circle has center at the origin and is tangent to all four corner circles.

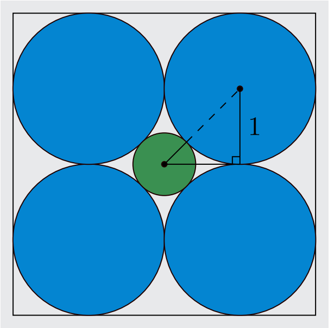


Figure: Filling 2D square with corner circles and center circle

Another example of high dimensional weirdness

In $\mathbb{R}^2$, consider the five depicted circles in the square $[-2,2]^2$. The four "corner" circles are tangent to (two) edges of the square and tangent to each other. Each of them has radius 1. The "center" circle has center at the origin and is tangent to all four corner circles.

The radius of the center circle is $\sqrt{2}-1 \approx 0.414$. Hence, it is smaller than each of the corner circles (as is visibly apparent).

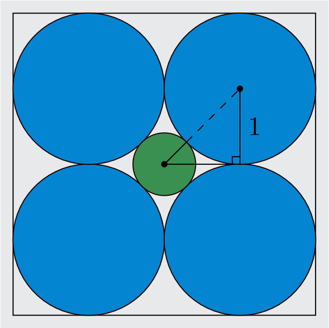


Figure: Filling 2D square with corner circles and center circle

Another example of high dimensional weirdness

In $\mathbb{R}^2$, consider the five depicted circles in the square $[-2,2]^2$. The four “corner” circles are tangent to (two) edges of the square and tangent to each other. Each of them has radius 1. The “center” circle has center at the origin and is tangent to all four corner circles.

The radius of the center circle is $\sqrt{2}-1 \approx 0.414$. Hence, it is smaller than each of the corner circles (as is visibly apparent).

Generalize this: the (hyper)cube $[-2,2]^d$ in $\mathbb{R}^d$. In general, there are 2^d corner spheres, each with radius 1. There is one center sphere, with the origin as its center (same as the hypercube) and which is tangent to all corner spheres.

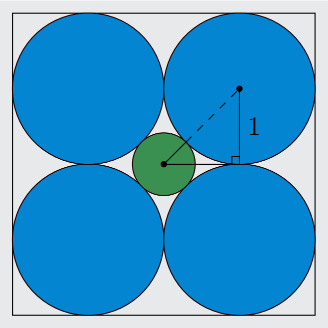


Figure: Filling 2D square with corner circles and center circle

Another example of high dimensional weirdness

As a consequence of the distance formula in $\mathbb{R}^d$, the radius of the center sphere is necessarily $\sqrt{d} - 1$.

Another example of high dimensional weirdness

As a consequence of the distance formula in $\mathbb{R}^d$, the radius of the center sphere is necessarily $\sqrt{d} - 1$. Note the consequences:

- When $d = 4$, then the center sphere is the same size as the corner spheres, since $\sqrt{4} - 1 = 1$.

Another example of high dimensional weirdness

As a consequence of the distance formula in $\mathbb{R}^d$, the radius of the center sphere is necessarily $\sqrt{d} - 1$. Note the consequences:

- When $d = 4$, then the center sphere is the same size as the corner spheres, since $\sqrt{4} - 1 = 1$.
- The center sphere is larger than the corner spheres when $d \geq 5$, and once $d = 9$ we have that the radius of the center sphere is 2. So, that center sphere intersects the boundary of the cube.

Another example of high dimensional weirdness

As a consequence of the distance formula in $\mathbb{R}^d$, the radius of the center sphere is necessarily $\sqrt{d} - 1$. Note the consequences:

- When $d = 4$, then the center sphere is the same size as the corner spheres, since $\sqrt{4} - 1 = 1$.
- The center sphere is larger than the corner spheres when $d \geq 5$, and once $d = 9$ we have that the radius of the center sphere is 2. So, that center sphere intersects the boundary of the cube.
- For $d \geq 10$, the center sphere contains points that are *outside of* the hypercube. (Despite still being tangent to all 2^d corner spheres, which “surround” it and are *entirely contained* within the hypercube.

Outline

k-means Clustering

We previously spent time on k-means clustering. Here is a quick recap for data $\mathcal{S} = \{\mathbf{x}_i\}_{i=1}^{n}$, with $\mathbf{x}_i \in \mathbb{R}^d$.

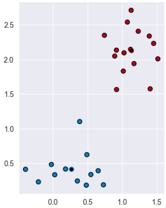


Figure: Result of k-means, 2 centroids in black

k-means Clustering

We previously spent time on k-means clustering. Here is a quick recap for data $\mathcal{S} = \{\mathbf{x}_i\}_{i=1}^n$, with $\mathbf{x}_i \in \mathbb{R}^d$.

You choose k, the number of clusters and randomly initialize k centroids $\mu_1, \mu_2, \ldots \mu_k$, each $\mu_i \in \mathbb{R}^d$. Clusters $C_1, C_2, \ldots, C_k$ are determined as follows.

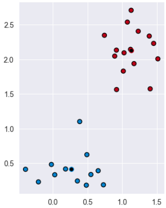


Figure: Result of k-means, 2 centroids in black

k-means Clustering

We previously spent time on k-means clustering. Here is a quick recap for data $\mathcal{S} = \{\mathbf{x}_i\}_{i=1}^n$, with $\mathbf{x}_i \in \mathbb{R}^d$.

You choose k, the number of clusters and randomly initialize k centroids $\mu_1, \mu_2, \ldots \mu_k$, each $\mu_i \in \mathbb{R}^d$. Clusters $C_1, C_2, \ldots, C_k$ are determined as follows.

1. For each data point $\mathbf{x}_i$, determine $j(i)$ with $1 \leq j(i) \leq k$, so that $\mu_{j(i)}$ is the closest centroid to $\mathbf{x}_i$. Then, $\mathbf{x}_i \in C_j$ precisely when $j = j(i)$.

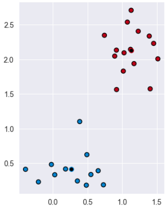


Figure: Result of k-means, 2 centroids in black

k-means Clustering

We previously spent time on k-means clustering. Here is a quick recap for data $\mathcal{S} = \{\mathbf{x}_i\}_{i=1}^n$, with $\mathbf{x}_i \in \mathbb{R}^d$.

You choose k, the number of clusters and randomly initialize k centroids $\mu_1, \mu_2, \ldots \mu_k$, each $\mu_i \in \mathbb{R}^d$. Clusters $C_1, C_2, \ldots, C_k$ are determined as follows.

1. For each data point $\mathbf{x}_i$, determine $j(i)$ with $1 \leq j(i) \leq k$, so that $\mu_{j(i)}$ is the closest centroid to $\mathbf{x}_i$. Then, $\mathbf{x}_i \in C_j$ precisely when $j = j(i)$.
2. Update $\mu_1, \mu_2, \ldots, \mu_k$ so that, for $1 \leq j \leq k$, the centroid of C_j is μ_j, i.e., $\mu_j = \frac{1}{|C_j|} \sum_{\mathbf{x}_i \in C_j} \mathbf{x}_i$.

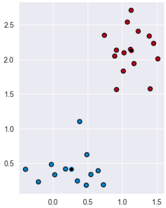


Figure: Result of k-means, 2 centroids in black

k-means Clustering

We previously spent time on k-means clustering. Here is a quick recap for data $\mathcal{S} = \{\mathbf{x}_i\}_{i=1}^n$, with $\mathbf{x}_i \in \mathbb{R}^d$.

You choose k, the number of clusters and randomly initialize k centroids $\mu_1, \mu_2, \dots \mu_k$, each $\mu_i \in \mathbb{R}^d$. Clusters $C_1, C_2, \dots, C_k$ are determined as follows.

1. For each data point $\mathbf{x}_i$, determine $j(i)$ with $1 \leq j(i) \leq k$, so that $\mu_{j(i)}$ is the closest centroid to $\mathbf{x}_i$. Then, $\mathbf{x}_i \in C_j$ precisely when $j = j(i)$.
2. Update $\mu_1, \mu_2, \dots, \mu_k$ so that, for $1 \leq j \leq k$, the centroid of C_j is μ_j, i.e., $\mu_j = \frac{1}{|C_j|} \sum_{\mathbf{x}_i \in C_j} \mathbf{x}_i$.
3. Iterate steps 1 and 2 until the assignment $i \mapsto j(i)$ that is made in 1 is the same as it was in the previous iteration.

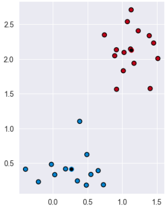


Figure: Result of k-means, 2 centroids in black

k-means Clustering

1. For each data point $\mathbf{x}_i$, determine $j(i)$ with $1 \leq j(i) \leq k$, so that $\mu_{j(i)}$ is the closest centroid to $\mathbf{x}_i$. Then, $\mathbf{x}_i \in C_j$ precisely when $j = j(i)$.
2. Update $\mu_1, \mu_2, \ldots, \mu_k$ so that, for $1 \leq j \leq k$, the centroid of C_j is μ_j, i.e., $\mu_j = \frac{1}{|C_j|} \sum_{\mathbf{x}_i \in C_j} \mathbf{x}_i$.
3. Iterate steps 1 and 2 until the assignment $i \mapsto j(i)$ that is made in 1 is the same as it was in the previous iteration.

In any iteration of 1, for $\mathbf{x}_i$ to change its cluster, it is necessary that $|\mathbf{x}_i - \mu_{j(i)}|$ decreases, so $\sum_{i=1}^{n} |\mathbf{x}_i - \mu_{j(i)}|^2$ decreases.

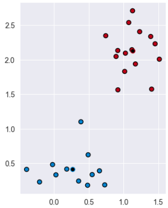


Figure: Result of k-means, 2 centroids in black

k-means Clustering

1. For each data point $\mathbf{x}_i$, determine $j(i)$ with $1 \leq j(i) \leq k$, so that $\mu_{j(i)}$ is the closest centroid to $\mathbf{x}_i$. Then, $\mathbf{x}_i \in C_j$ precisely when $j = j(i)$.
2. Update $\mu_1, \mu_2, \ldots, \mu_k$ so that, for $1 \leq j \leq k$, the centroid of C_j is μ_j, i.e., $\mu_j = \frac{1}{|C_j|} \sum_{\mathbf{x}_i \in C_j} \mathbf{x}_i$.
3. Iterate steps 1 and 2 until the assignment $i \mapsto j(i)$ that is made in 1 is the same as it was in the previous iteration.

In any iteration of 1, for $\mathbf{x}_i$ to change its cluster, it is necessary that $|\mathbf{x}_i - \mu_{j(i)}|$ decreases, so $\sum_{i=1}^{n} |\mathbf{x}_i - \mu_{j(i)}|^2$ decreases. In any iteration of 2, setting $\mu_j = \frac{1}{|C_j|} \sum_{\mathbf{x}_i \in C_j} \mathbf{x}_i$ will minimize $\sum_{\mathbf{x}_i \in C_j} |\mathbf{x}_i - \mu_j|^2$, and so $\sum_{i=1}^{n} |\mathbf{x}_i - \mu_{j(i)}|^2$ decreases (or stays the same) on this step.

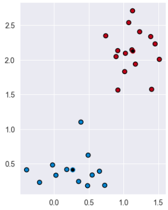


Figure: Result of k-means, 2 centroids in black

k-means Clustering

1. For each data point $\mathbf{x}_i$, determine $j(i)$ with $1 \leq j(i) \leq k$, so that $\mu_{j(i)}$ is the closest centroid to $\mathbf{x}_i$. Then, $\mathbf{x}_i \in C_j$ precisely when $j = j(i)$.
2. Update $\mu_1, \mu_2, \ldots, \mu_k$ so that, for $1 \leq j \leq k$, the centroid of C_j is μ_j, i.e., $\mu_j = \frac{1}{|C_j|} \sum_{\mathbf{x}_i \in C_j} \mathbf{x}_i$.
3. Iterate steps 1 and 2 until the assignment $i \mapsto j(i)$ that is made in 1 is the same as it was in the previous iteration.

In any iteration of 1, for $\mathbf{x}_i$ to change its cluster, it is necessary that $|\mathbf{x}_i - \mu_{j(i)}|$ decreases, so $\sum_{i=1}^{n} |\mathbf{x}_i - \mu_{j(i)}|^2$ decreases. In any iteration of 2, setting $\mu_j = \frac{1}{|C_j|} \sum_{\mathbf{x}_i \in C_j} \mathbf{x}_i$ will minimize $\sum_{\mathbf{x}_i \in C_j} |\mathbf{x}_i - \mu_j|^2$, and so $\sum_{i=1}^{n} |\mathbf{x}_i - \mu_{j(i)}|^2$ decreases (or stays the same) on this step.
Thus, the algorithm terminates: there are finitely many points in $\mathcal{S}$, so there are only a finite number of possibilities for the list $\mu_1, \mu_2, \ldots, \mu_k$.

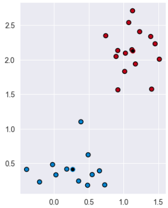


Figure: Result of k-means, 2 centroids in black

DBSCAN Clustering

DBSCAN (Density-Based Spatial Clustering for Applications with Noise) does not require you to choose a number of clusters at the start.

DBSCAN Clustering

DBSCAN (Density-Based Spatial Clustering for Applications with Noise) does not require you to choose a number of clusters at the start.
There are two important hyperparameters that you choose:

- ε (`eps` in scikit-learn, a certain radius in the procedure); and

DBSCAN Clustering

DBSCAN (Density-Based Spatial Clustering for Applications with Noise) does not require you to choose a number of clusters at the start.
There are two important hyperparameters that you choose:

- ε (`eps` in scikit-learn, a certain radius in the procedure); and
- `minPts` (`min_samples` in scikit-learn).

DBSCAN Clustering

DBSCAN (Density-Based Spatial Clustering for Applications with Noise) does not require you to choose a number of clusters at the start.
There are two important hyperparameters that you choose:

- ε (`eps` in scikit-learn, a certain radius in the procedure); and
- `minPts` (`min_samples` in scikit-learn).

Describing how clusters are formed requires some terminology.

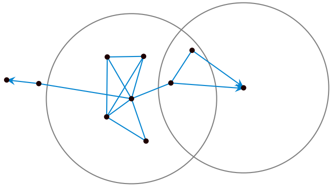


Figure: Points reachable from core point, `minPts`$= 4$

DBSCAN Clustering

DBSCAN (Density-Based Spatial Clustering for Applications with Noise) does not require you to choose a number of clusters at the start.
There are two important hyperparameters that you choose:

- ε (`eps` in scikit-learn, a certain radius in the procedure); and
- `minPts` (`min_samples` in scikit-learn).

Describing how clusters are formed requires some terminology.
As with k-means, the distance function (metric) that is used is a central part of the process. Unlike the centroids used in k-means, though, a different metric would not require a change to the procedure (other than different distance computations).

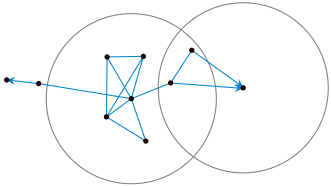


Figure: Points reachable from core point, `minPts`$= 4$

DBSCAN Clustering

Have (training) data $\mathcal{S} \subset \mathbb{R}^d$ (no label given for an $\mathbf{x} \in \mathcal{S}$). Write $d(\mathbf{p}, \mathbf{q})$ for the distance between points **p** and **q**, which could be distance in any metric.

DBSCAN Clustering

Have (training) data $\mathcal{S} \subset \mathbb{R}^d$ (no label given for an $\mathbf{x} \in \mathcal{S}$). Write $d(\mathbf{p}, \mathbf{q})$ for the distance between points $\mathbf{p}$ and $\mathbf{q}$, which could be distance in any metric.

- Any $\mathbf{p} \in \mathcal{S}$ is a **core point** if `minPts` training samples are within ε of it; so,

$$\#\{\mathbf{x} \in \mathcal{S} \mid d(\mathbf{p}, \mathbf{x}) < \varepsilon\} \geq \texttt{minPts}.$$

DBSCAN Clustering

Have (training) data $\mathcal{S} \subset \mathbb{R}^d$ (no label given for an $\mathbf{x} \in \mathcal{S}$). Write $d(\mathbf{p}, \mathbf{q})$ for the distance between points $\mathbf{p}$ and $\mathbf{q}$, which could be distance in any metric.

- Any $\mathbf{p} \in \mathcal{S}$ is a **core point** if `minPts` training samples are within ε of it; so,

$$\#\{\mathbf{x} \in \mathcal{S} \mid d(\mathbf{p}, \mathbf{x}) < \varepsilon\} \geq \texttt{minPts}.$$

- Given core point $\mathbf{p}$, and $\mathbf{x} \in \mathcal{S}$, $\mathbf{x}$ is **directly reachable** from $\mathbf{p}$ if $d(\mathbf{p}, \mathbf{x}) < \varepsilon$.

DBSCAN Clustering

Have (training) data $\mathcal{S} \subset \mathbb{R}^d$ (no label given for an $\mathbf{x} \in \mathcal{S}$). Write $d(\mathbf{p}, \mathbf{q})$ for the distance between points $\mathbf{p}$ and $\mathbf{q}$, which could be distance in any metric.

- Any $\mathbf{p} \in \mathcal{S}$ is a **core point** if `minPts` training samples are within ε of it; so,

$$\#\{\mathbf{x} \in \mathcal{S} \mid d(\mathbf{p}, \mathbf{x}) < \varepsilon\} \geq \texttt{minPts}.$$

- Given core point $\mathbf{p}$, and $\mathbf{x} \in \mathcal{S}$, $\mathbf{x}$ is **directly reachable** from $\mathbf{p}$ if $d(\mathbf{p}, \mathbf{x}) < \varepsilon$.
- Given core point $\mathbf{p}$, and $\mathbf{x} \in \mathcal{S}$, $\mathbf{x}$ is **reachable** from $\mathbf{p}$ if there exist $\mathbf{p}_1, \ldots, \mathbf{p}_m$ such that $\mathbf{p}_{i+1}$ is directly reachable from $\mathbf{p}_i$, all $1 \leq i \leq m-1$, and $\mathbf{p}_1 = \mathbf{p}$, $\mathbf{p}_m = \mathbf{x}$.

DBSCAN Clustering

Have (training) data $\mathcal{S} \subset \mathbb{R}^d$ (no label given for an $\mathbf{x} \in \mathcal{S}$). Write $d(\mathbf{p}, \mathbf{q})$ for the distance between points $\mathbf{p}$ and $\mathbf{q}$, which could be distance in any metric.

- Any $\mathbf{p} \in \mathcal{S}$ is a **core point** if `minPts` training samples are within ε of it; so,

$$\#\{\mathbf{x} \in \mathcal{S} \mid d(\mathbf{p}, \mathbf{x}) < \varepsilon\} \geq \texttt{minPts}.$$

- Given core point $\mathbf{p}$, and $\mathbf{x} \in \mathcal{S}$, $\mathbf{x}$ is **directly reachable** from $\mathbf{p}$ if $d(\mathbf{p}, \mathbf{x}) < \varepsilon$.
- Given core point $\mathbf{p}$, and $\mathbf{x} \in \mathcal{S}$, $\mathbf{x}$ is **reachable** from $\mathbf{p}$ if there exist $\mathbf{p}_1, \ldots, \mathbf{p}_m$ such that $\mathbf{p}_{i+1}$ is directly reachable from $\mathbf{p}_i$, all $1 \leq i \leq m-1$, and $\mathbf{p}_1 = \mathbf{p}$, $\mathbf{p}_m = \mathbf{x}$.

With this terminology, let $\mathbf{p} \in \mathcal{S}$ be a core point. The cluster, $C_{\mathbf{p}}$ say, is the set of all points in $\mathcal{S}$ (including $\mathbf{p}$) that are reachable from $\mathbf{p}$.

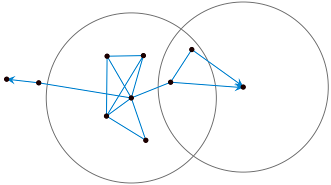


Figure: Points reachable from core point, `minPts`$= 4$

DBSCAN Clustering

Have (training) data $\mathcal{S} \subset \mathbb{R}^d$ (no label given for an $\mathbf{x} \in \mathcal{S}$). Write $d(\mathbf{p}, \mathbf{q})$ for the distance between points $\mathbf{p}$ and $\mathbf{q}$, which could be distance in any metric.

- Any $\mathbf{p} \in \mathcal{S}$ is a **core point** if `minPts` training samples are within ε of it; so,

$$\#\{\mathbf{x} \in \mathcal{S} \mid d(\mathbf{p}, \mathbf{x}) < \varepsilon\} \geq \texttt{minPts}.$$

- Given core point $\mathbf{p}$, and $\mathbf{x} \in \mathcal{S}$, $\mathbf{x}$ is **directly reachable** from $\mathbf{p}$ if $d(\mathbf{p}, \mathbf{x}) < \varepsilon$.
- Given core point $\mathbf{p}$, and $\mathbf{x} \in \mathcal{S}$, $\mathbf{x}$ is **reachable** from $\mathbf{p}$ if there exist $\mathbf{p}_1, \ldots, \mathbf{p}_m$ such that $\mathbf{p}_{i+1}$ is directly reachable from $\mathbf{p}_i$, all $1 \leq i \leq m - 1$, and $\mathbf{p}_1 = \mathbf{p}$, $\mathbf{p}_m = \mathbf{x}$.

With this terminology, let $\mathbf{p} \in \mathcal{S}$ be a core point. The cluster, $C_{\mathbf{p}}$ say, is the set of all points in $\mathcal{S}$ (including $\mathbf{p}$) that are reachable from $\mathbf{p}$. A point in $C_{\mathbf{p}}$ that is not a core point might be considered the "edge" of the cluster, as no point may be reached from it.

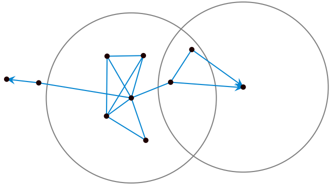


Figure: Points reachable from core point, `minPts`$= 4$