# Deadlines

<table>
<thead>
<tr>
<th>Assignment</th>
<th>Date of assignment</th>
<th>Deadline (midnight 23:59)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HW1</td>
<td>Sep 5</td>
<td>Sep 18</td>
</tr>
<tr>
<td>HW2</td>
<td>Sep 19</td>
<td>Oct 2</td>
</tr>
<tr>
<td>HW3</td>
<td>Oct 3</td>
<td>Oct 16</td>
</tr>
<tr>
<td>Paper summary</td>
<td>Oct 10</td>
<td>Oct 23</td>
</tr>
<tr>
<td>HW4</td>
<td>Oct 17</td>
<td>Oct 30</td>
</tr>
<tr>
<td>HW5</td>
<td>Oct 31</td>
<td>Nov 13</td>
</tr>
<tr>
<td>HW6</td>
<td>Nov 21</td>
<td>Dec 4</td>
</tr>
<tr>
<td>Project</td>
<td>TBA</td>
<td>Dec 12 - 14</td>
</tr>
</tbody>
</table>

*All deadlines are subject to change, check out Slack and website for updates*
Now you are ready for real-life tasks.

Overview of ML

Supervised Learning
Unsupervised Learning
Performance metrics
Deep Learning
Ensemble learning
Other

Nearest Neighbour Classifier
Linear regression
Decision trees
Overfitting
Train/val split
Cross Validation
Algorithm
Model selection
Model implementation
Data preprocessing
Classification vs regression
Hierarchical clustering
DBSCAN
PCA
K-means
T-SNE
UMAP

Accuracy
MSE & RMSE
Recall, precision
F1-score
Confusion matrix
ROC & AUC

Guest talk
Concluding remarks
Final presentations

HW
Six homeworks
(10 points each)

R
Paper review
(15 points)
High Dimensional Space

Low Dimensional Space

Dimensionality reduction

Unsupervised Learning

Regression

Supervised Learning

Classification

Reinforcement Learning

Machine Learning

Torus image credit: https://mathematica.stackexchange.com/questions/39879/create-a-torus-with-a-hexagonal-mesh-for-3d-printing
What is the problem with high-dimensional things?

- **Hard to visualise**
- Methods trained on high-dimensional data suffer from the **curse of dimensionality**
- **Algorithms** tend to get slow
Compute covariance matrix

Perform eigen decomposition

Compute new coordinates
**Supervised Learning pipeline**

1. **200D raw data**
2. **Train/test split**
3. **Normalisation (subtract mean)**
4. **PCA**
   - Keep few PCs (90% variance)
5. **Find the best model using CV**
6. **Evaluate final model on the test set**

**Safe place**

**Profit**
Figure 6: A comparison between UMAP and t-SNE projections of a 3D woolly mammoth skeleton (50,000 points) into 2 dimensions, with various settings for parameters. Notice how much more global structure is preserved with UMAP, particularly with larger values of `n_neighbors`.

UMAP explained and compared to t-SNE: https://pair-code.github.io/understanding-umap/
Machine Learning

Reinforcement Learning

Classification

Regression

Supervised Learning

Unsupervised Learning

Clustering

Dimensionality reduction
Machine Learning

- Classification
- Regression
- Supervised Learning
- Reinforcement Learning

Unsupervised Learning
- Dimensionality reduction
- Clustering
Unlabelled data

Dimensionality reduction

Unsupervised Learning

Clustering
Clustering methods are looking for structure in unlabeled data.
How come stuff does not have annotations?
(Why unsupervised learning is a thing?)
How come stuff does not have annotations?
(Why unsupervised learning is a thing?)

Labels are noisy and not trustworthy
How come stuff does not have annotations?
(Why unsupervised learning is a thing?)

Labels are noisy and not trustworthy

Labels are super expensive and laborious
How come stuff does not have annotations? (Why unsupervised learning is a thing?)

Labels are noisy and not trustworthy

Labels simply do not exist

Labels are super expensive and laborious
How come stuff does not have annotations?
(Why unsupervised learning is a thing?)

Labels are noisy and not trustworthy

Labels are super expensive and laborious

Labels simply do not exist

We learn with very limited or no supervision
Two groups labeled by an expert
No labels. Are there any meaningful groups?
Hierarchical clustering

K-means

DBSCAN

centroids

noise
Hierarchical clustering
Hierarchical clustering

At first all points are considered to be separate clusters
Hierarchical clustering

Two closest clusters are merged into one
Hierarchical clustering

Two closest clusters are merged into one

Euclidean distance is calculated between all clusters
Hierarchical clustering

Two closest clusters are merged into one

Euclidean distance is calculated between all clusters
Hierarchical clustering

Two closest clusters are merged into one

Euclidean distance is calculated between all clusters
Hierarchical clustering

Two closest clusters are merged into one

These points are now belong to one cluster
Hierarchical clustering

Two closest clusters are merged into one.

Euclidean distance is calculated between all clusters.
Hierarchical clustering

Now we have to **re-calculate distances** between clusters again to figure out next pair of **clusters to merge**.
When distance is calculated between single point cluster and multiple point cluster few options exist
Hierarchical clustering

When distance is calculated between **single point** cluster and **multiple point cluster** few options exist.

I. Distance between **closest points** defines the distance between clusters (i.e. **single linkage**).
Hierarchical clustering

When distance is calculated between **single point** cluster and **multiple point cluster** few options exist.

**I.** Distance between closest points defines the distance between clusters (i.e., **single linkage**).

**II.** Distance between furthest points defines the distance between clusters (i.e., **complete linkage**).
When distance is calculated between single point cluster and multiple point cluster few options exist:

I. Distance between closest points defines the distance between clusters (i.e. single linkage).

II. Distance between furthest points defines the distance between clusters (i.e. complete linkage).

III. Distance between averages defines the distance between clusters (i.e. average linkage).
Hierarchical clustering

Two closest clusters are merged into one

Euclidean distance is calculated between all clusters

*assume we use single linkage
Hierarchical clustering

Two closest clusters are merged into one

Euclidean distance is calculated between all clusters

*assume we use single linkage
Hierarchical clustering

According to **single linkage** distance between any cluster and the new **cluster** is the distance between closest points of these clusters.

*assume we use **single linkage**
Hierarchical clustering

Two closest clusters are merged into one

Euclidean distance is calculated between all clusters

*assume we use single linkage
Hierarchical clustering

Two closest clusters are merged into one

Euclidean distance is calculated between all clusters

*assume we use single linkage
Hierarchical clustering

Two closest clusters are merged into one

Euclidean distance is calculated between all clusters

*assume we use single linkage
Hierarchical clustering

Two closest clusters are merged into one.

Euclidean distance is calculated between all clusters.

*Assume we use single linkage*
Hierarchical clustering

Two **closest** clusters are **merged** into one

Euclidean distance is calculated between all clusters

*assume we use **single linkage**
Hierarchical clustering

Two closest clusters are merged into one

Euclidean distance is calculated between all clusters

*assume we use single linkage
Hierarchical clustering

Two **closest** clusters are **merged** into one

**Euclidean** distance is calculated between all clusters

*assume we use **single linkage***
Hierarchical clustering

Two closest clusters are merged into one

Euclidean distance is calculated between all clusters

*assume we use single linkage
Hierarchical clustering

Two closest clusters are merged into one

Euclidean distance is calculated between all clusters

*assume we use single linkage*
Hierarchical clustering

Two closest clusters are merged into one

Euclidean distance is calculated between all clusters

*assume we use single linkage
Hierarchical clustering

Two **closest** clusters are **merged** into one

**Euclidean** distance is calculated between all clusters

*assume we use **single linkage***
Hierarchical clustering

Two closest clusters are merged into one

Euclidean distance is calculated between all clusters

*assume we use single linkage
Hierarchical clustering

Two closest clusters are merged into one

Euclidean distance is calculated between all clusters

*assume we use single linkage
Hierarchical clustering
Hierarchical clustering
Hierarchical clustering

Dendrogram
Hierarchical clustering
Hierarchical clustering

Distance

Clusters

Dendrogram
Hierarchical clustering
Hierarchical clustering
Hierarchical clustering
Hierarchical clustering

Distance

Clusters

Dendrogram

Dendrogram

Distance vs Clusters
Hierarchical clustering

Distance

Clusters

Dendrogram
Hierarchical clustering

Why are they on the same height?
Hierarchical clustering

Because **distances** between clusters are the same!

Why are they on the same height?

Because distances between clusters are the same!
Hierarchical clustering

Distance

Clusters

Dendrogram
Hierarchical clustering

Distance

Clusters

Dendrogram
Hierarchical clustering

These are also the same!
Hierarchical clustering

Dendrogram
Hierarchical clustering

Distance

Clusters

Dendrogram
Hierarchical clustering

Dendrogram

Distance

Clusters
Hierarchical clustering

Dendrogram

Clusters

Distance

2 0 1 5 3 4 6
Hierarchical clustering

What is it useful for?
Hierarchical clustering

We can **cut it**!

What is it useful for?
Hierarchical clustering

We can **cut it**!
Give me **two** clusters!

What is it useful for?
Hierarchical clustering

What is it useful for?

We can **cut it**!
Give me **two** clusters!

What is it useful for?
Hierarchical clustering

We can cut it!
Give me two clusters!

Here are your two clusters

What is it useful for?
Hierarchical clustering

We can **cut it!**
Give me **three** clusters!

What is it useful for?
Hierarchical clustering

What is it useful for?

We can **cut it**!
Give me **three** clusters!

What is it useful for?
Hierarchical clustering

We can **cut it**!
Give me **three** clusters!

What is it useful for?
Hierarchical clustering

Here are your three clusters

We can cut it! Give me three clusters!

What is it useful for?
Hierarchical clustering

What is your take on?

What is it useful for?

Distance

Clusters

What is it useful for?
Hierarchical clustering

Dendrogram is a way to store a structure of our data

What is it useful for?
At first all points are considered to be separate clusters.

Euclidean distance is calculated between all clusters.

Repeat as many times as there are clusters.

Until there is one cluster left.

Two closest clusters are merged into one.
At first **all points** are considered to be **separate clusters**

Euclidean distance is calculated between **all clusters**

Repeat as many times as there are clusters

**Agglomerative strategy**

Until there is **one cluster** left

Two **closest** clusters are **merged** into one
At first all points are considered to be separate clusters.

Euclidean distance is calculated between all clusters.

What is the complexity?

Repeat as many times as there are clusters.

Until there is one cluster left.

Two closest clusters are merged into one.
At first all points are considered to be separate clusters. Euclidean distance is calculated between all clusters. Repeat as many times as there are clusters. What is the complexity? O(N^3)

Until there is one cluster left. Two closest clusters are merged into one.
Hierarchical clustering is super popular for small/medium size datasets.
At first all points are considered to be one cluster.

Divisive strategy:

- All points are initially considered to be one cluster.
- Use some other clustering algorithm (e.g., K-Means) to split clusters as many times as there are clusters.
- Repeat until all points are considered separate clusters.

Until there all points are considered separate clusters.
Hierarchical clustering

K-means

centroids

DBSCAN

noise
Hierarchical clustering

K-means

centroids

DBSCAN

noise
K-means clustering

All points are considered to be clusterless at first.
K-means clustering

Randomly spawn $K$ potential cluster centres
K-means clustering

Randomly spawn $K$ potential cluster centres (multiple possibilities)
K-means clustering

Randomly spawn $K$ potential cluster centres (multiple possibilities)
K-means clustering

Randomly spawn \( K \) potential cluster centres (multiple possibilities)

Choose centroids randomly within dataset ranges
K-means clustering

Randomly spawn $K$ potential cluster centres (multiple possibilities)
K-means clustering

Randomly spawn K potential cluster centres (multiple possibilities)

Random cases from dataset are selected as centroids
K-means clustering

Random cases from dataset are selected as centroids

Randomly spawn $K$ potential cluster centres (multiple possibilities)
Randomly spawn $K$ potential cluster centres (multiple possibilities)

There are more options… (methods of initializing K-means clustering)

https://stats.stackexchange.com/questions/317493/methods-of-initializing-k-means-clustering
K-means clustering

Randomly spawn $K$ potential cluster centres (multiple possibilities)

Choose centroids randomly within dataset ranges
K-means clustering

Randomly spawn \( K \) potential cluster centres (multiple possibilities)

New centroids (cluster centres)
Now as centroids were assigned we can start iterating K-means algorithm
K-means clustering

Compute **distances** from all points to new centroids
K-means clustering

Assign each point to the closest centroid

Compute distances from all points to new centroids
K-means clustering

Assign each point to the closest centroid

Compute distances from all points to new centroids
K-means clustering

Assign each point to the closest centroid

Compute distances from all points to new centroids
K-means clustering

Assign each point to the closest centroid

Compute distances from all points to new centroids
K-means clustering

Assign each point to the closest centroid

Compute distances from all points to new centroids

What about this case?
K-means clustering

1. **Compute distances** from all points to new centroids.
2. **Assign each point** to the **closest centroid**.
3. **Break the tie** by random choice.
K-means clustering

Compute distances from all points to new centroids

Assign each point to the closest centroid

Break the tie by random choice
K-means clustering

Assign each point to the closest centroid

Compute distances from all points to new centroids
K-means clustering

Assign each point to the closest centroid

Compute distances from all points to new centroids
K-means clustering
These are our clusters so far…

Cluster #1

Cluster #2
K-means clustering

Recalculate centroids for each cluster
K-means clustering

Recalculate centroids for each cluster
K-means clustering

Recalculate centroids for each cluster

$[2, 3, 5]$
K-means clustering

Recalculate centroids for each cluster

$[2, 3, 5]$

$\bar{x} = 3.33$
K-means clustering

Recalculate centroids for each cluster

\[ [2, 3, 5] \quad [5, 6, 7] \]

\[ \bar{x} = 3.33 \quad \bar{y} = ? \]
K-means clustering

Recalculate centroids for each cluster

\[[2, 3, 5]\] \[\bar{x} = 3.33\]
\[[5, 6, 7]\] \[\bar{y} = 6\]
K-means clustering

Recalculate centroids for each cluster

\[ \bar{x} = 3.33 \quad \bar{y} = 6 \]

New centroid for blue cluster

[2, 3, 5] \quad [5, 6, 7]
K-means clustering

Recalculate centroids for each cluster

\[ [3, 6, 7, 7] \]

\[ \bar{x} = 5.75 \]
K-means clustering

Recalculate centroids for each cluster

$[3, 6, 7, 7]$ \hspace{1cm} $[4, 6, 6, 8]$

$\bar{x} = 5.75$ \hspace{1cm} $\bar{y} = 6$
K-means clustering

Recalculate centroids for each cluster

\[ \bar{x} = 5.75 \quad \bar{y} = 6 \]

[3, 6, 7, 7] \quad [4, 6, 6, 8]
K-means clustering

Recalculate centroids for each cluster

\[ [3, 6, 7, 7] \quad [4, 6, 6, 8] \]

\[ \bar{x} = 5.75 \quad \bar{y} = 6 \]
K-means clustering

Now as centroids were assigned we can start iterating K-means algorithm
K-means clustering

Compute distances from all points to new centroids
K-means clustering
K-means clustering

These are our clusters so far…
K-means clustering

Recalculate **centroids** for each cluster
K-means clustering

Recalculate **centroids** for each cluster
K-means clustering

Recalculate **centroids** for each cluster.
K-means clustering

Rinse and repeat until centroids are stable within certain radius (epsilon)
K-means clustering

Rinse and repeat until centroids *are stable* within certain radius (epsilon)
K-means clustering

Rinse and repeat until centroids are stable within certain radius (epsilon)
K-means clustering

Rinse and repeat until centroids are stable within a certain radius (epsilon)
K-means clustering

Rinse and repeat until centroids are stable within certain radius (epsilon)
K-means clustering

Rinse and repeat until centroids are stable within a certain radius (epsilon)
K-means clustering

Online visualisation: https://www.naftaliharris.com/blog/visualizing-k-means-clustering/
Randomly initialise \( K \) cluster centres

*e.g. \( K = 2 \)*

**K-means** algorithm
Randomly initialise $K$ cluster centres

Assign all points to closest cluster centres

*K-means* algorithm

*e.g. $K = 2$*
Randomly initialise $K$ cluster centres

Assign all points to closest cluster centres

*K-means* algorithm

For each newly formed cluster recompute its centre
Randomly initialise $K$ cluster centres

Assign all points to closest cluster centres

Repeat until the centres are not stable within predefined radius

**K-means algorithm**

For each newly formed cluster recompute its centre
**K-means algorithm**

What is the complexity?
Randomly initialise $K$ cluster centres

Assign all points to closest cluster centres

*e.g. $K = 2$

$K$-means algorithm

$K$-means is computationally difficult (NP-hard)*

but it has been shown to converge in finite number of steps

For each newly formed cluster recompute its centre

Repeat until the centres are not stable within predefined radius

*https://cseweb.ucsd.edu/~avattani/papers/kmeans_hardness.pdf
In both algorithms, $K$ plays an important role.
Any good way to choose K?
Any good way to choose $K$?

The rule of thumb is to choose $\sqrt{n/2}$ as $K$.
Any good way to choose $K$?

The rule of thumb is to choose $\sqrt{n/2}$ as $K$.

**Elbow method**: increase $K$ until it does not help to describe data better.
Elbow method to predict K

Elbow method for K-means: https://bl.ocks.org/rpgove/0060ff3b656618e9136b
Elbow method to predict K

Elbow method for K-means: https://bl.ocks.org/rpgove/0060ff3b656618e9136b
Elbow method to predict $K$

$K = 1$

$K = 2$

$K = 3$

Elbow method for K-means: https://bl.ocks.org/rpgove/0060ff3b656618e9136b
Elbow method to predict $K$

For each case we calculate **sum of squared errors** (SSE)

Elbow method for K-means: https://bl.ocks.org/rpgove/0060ff3b656618e9136b
Elbow method to predict $K$

For each potential number $K$ we calculate **sum of squared Euclidian distances to the closest centroid**

Elbow method for K-means: https://bl.ocks.org/rpgove/0060ff3b656618e9136b
Elbow method to predict $K$

For each potential number $K$ we calculate **sum of squared Euclidian distances** to the closest centroid.

$$SSE = \sum_{j=1}^{k} \sum_{i=1}^{n} d(c_j, p_i)^2$$

Elbow method for K-means: https://bl.ocks.org/rpgove/0060ff3b656618e9136b
Elbow method to predict K

For each potential number K we calculate **sum of squared Euclidian distances** to the closest centroid.

\[ SSE = \sum_{j=1}^{k} \sum_{i=1}^{n} d(c_j, p_i)^2 \]

where:
- \( k \) is the number of clusters
- \( n \) is the number of points in the \( j \)th cluster
- \( d(c_j, p_i) \) is the Euclidean distance from the centroid of the \( j \)th cluster to the \( i \)th point in the \( j \)th cluster

Elbow method for K-means: https://bl.ocks.org/rpgove/0060ff3b656618e9136b
Elbow method to predict K

Elbow method for K-means: https://bl.ocks.org/rpgove/0060ff3b656618e9136b

SSE: 35.42

SSE: 8.16

SSE: 5.33
Elbow method to predict K

Elbow method for K-means: https://bl.ocks.org/rpgove/0060ff3b656618e9136b
Elbow method to predict K

K = 1

K = 2

K = 3

SSE: 35.42
SSE: 8.16
SSE: 5.33

Elbow method for K-means: https://bl.ocks.org/rpgove/0060ff3b656618e9136b
Elbow method to predict K

Elbow method for K-means: https://bl.ocks.org/rpgove/0060ff3b656618e9136b
Elbow method to predict K

SSE: 35.42

SSE: 8.16

SSE: 5.33

Elbow method for K-means: https://bl.ocks.org/rpgove/0060ff3b656618e9136b
Silhouette method to predict $K$
Silhouette method to predict $K$
Silhouette method to predict $K$

For each point $p$
calculate: $a(p)$, $b(p)$ and $s(p)$
Silhouette method to predict $K$

For each point $p$ calculate:
- $a(p)$, average distance to points that belong to the same cluster as $p$.
Silhouette method to predict $K$

For each point $p$ calculate: $a(p)$, $b(p)$ and $s(p)$

- $a(p)$ - average distance to points that belong to the same cluster as $p$
- $b(p)$ - minimum average distance to clusters that $p$ does not belong to
**Silhouette** method to predict $K$

For each point $p$
- calculate: $a(p)$ and $b(p)$

$a(p)$ - average distance to points that belong to the same cluster as $p$.

$b(p)$ - minimum average distance to clusters that $p$ does not belong to.
Silhouette method to predict $K$

For each point $p$ calculate:

- $a(p)$ - average distance to points that belong to the same cluster as $p$

- $b(p)$ - minimum average distance to clusters that $p$ does not belong to

$K = 3$

$p$
**Silhouette** method to predict \( K \)

For each point \( p \) calculate: \( a(p) \) and \( b(p) \)

- \( a(p) \) - average distance to points that belong to the same cluster as \( p \)
- \( b(p) \) - minimum average distance to clusters that \( p \) does not belong to
**Silhouette** method to predict $K$

For each point $p$
calculate:  
$a(p)$ and $b(p)$

$a(p)$ - average distance to points that belong to the same cluster as $p$

$b(p)$ - minimum average distance to clusters that $p$ does not belong to

Average distance from $p$ to **green cluster** is 0.61

Average distance from $p$ to **red cluster** is 0.58

$K = 3$
Silhouette method to predict $K$

For each point $p$ calculate: $a(p)$ and $b(p)$

- $a(p)$ - average distance to points that belong to the same cluster as $p$
- $b(p)$ - minimum average distance to clusters that $p$ does not belong to

$K = 3$

Average distance from $p$ to green cluster is 0.61

Average distance from $p$ to red cluster is 0.58

In this case $b(p)$ is 0.58
**Silhouette** method to predict $K$

For each point $p$ calculate: $a(p)$, $b(p)$ and $s(p)$

- $a(p)$ - average distance to points that belong to the same cluster as $p$
- $b(p)$ - minimum average distance to clusters that $p$ does not belong to
**Silhouette** method to predict $K$

For each point $p$ calculate: $a(p)$, $b(p)$ and $s(p)$

- $a(p)$ - average distance to points that belong to the same cluster as $p$
- $b(p)$ - minimum average distance to clusters that $p$ does not belong to

$$s(p) = \frac{b(p) - a(p)}{\max\{a(p), b(p)\}}$$

**Silhouette score** for point $p$
Silhouette method to predict $K$

Overall silhouette score is an average of silhouette scores for all points.

For each point $p$ calculate: $a(p)$, $b(p)$ and $s(p)$.
Silhouette method to predict $K$

For each point $p$ calculate: $a(p)$, $b(p)$ and $s(p)$

Overall silhouette score is an average of silhouette scores for all points

The higher the total silhouette score, the better clustering
The rule of thumb is to choose $\sqrt{n/2}$ as $K$.

**Elbow method**: increase $K$ until it does not help to describe data better.

Any good way to choose $K$?

Overall silhouette score is an average of silhouette scores for all points.
Any good way to choose $K$?

No, not really.

The rule of thumb is to choose $\sqrt{n/2}$ as $K$.

Elbow method: increase $K$ until it does not help to describe data better.

Overall silhouette score is an average of silhouette scores for all points.
DBSCAN

Hierarchical clustering

K-means

centroids

DBSCAN

Does not require K
Density-based spatial clustering of applications with noise (DBSCAN)
DBSCAN

Two important parameters:
Two important parameters:

- \( \epsilon \) (epsilon), the radius of the neighbourhood, which is searched for each point.
Two important parameters:

- **epsilon**: Radius of the neighbourhood, which is searched for each point.
- **minPoints**: Minimum number of points in the neighbourhood to be in cluster.
Pick a random point
DBSCAN

Pick a random point

Start

minPoints = 3

epsilon
DBSCAN

Pick a random point

Search its epsilon neighbourhood

minPoints = 3

epsilon

Start
Pick a random point

If in neighbourhood >= minPoints points, this is cluster

minPoints = 3

epsilon
If in \textit{neighbourhood} \(>= \text{minPoints}\) points, this is \textit{cluster} (including itself)

\texttt{DBSCAN}
If in *neighbourhood* \( \geq \text{minPoints} \) points, this is *cluster* (including itself).

**DBSCAN**

\( \text{minPoints} = 3 \)

\( \text{epsilon} \)
DBSCAN

- Start point
- minPoints = 3
- epsilon

Move to the next point in the neighbourhood
Move to the next point in the neighbourhood

$\text{epsilon}$

minPoints = 3

DBSCAN
DBSCAN

Move to the next point in the neighbourhood

minPoints = 3
epsilon
DBSCAN

Move to the next point in the neighbourhood

minPoints = 3

epsilon
Move to the next point in the neighbourhood

\[ \text{minPoints} = 3 \]

\[ \text{epsilon} \]
If no points left in the neighbourhood, pick another random point.
DBSCAN

\[ \text{minPoints} = 3 \]

\[ \text{epsilon} \]
DBSCAN

minPoints = 3

epsilon
DBSCAN

minPoints = 3

epsilon
This point does not have enough points, but it is reachable from the point that belongs to cluster minPoints = 3
This point does not have enough points, but it is reachable from the point that belongs to cluster.
DBSCAN

minPoints = 3

epsilon
DBSCAN

minPoints = 3

epsilon

epsilon
Lonely points, are considered noise by DBSCAN.

minPoints = 3

epsilon

DBSCAN
Very good visualisation: https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/
DBSCAN

Hierarchical clustering

Parameters: **linkage type** and the **distance**

K-means

Parameters: **K** and the **distance**

DBSCAN

Parameters: **minPoints** and **epsilon**
Which of these **three** we should use and when?
Hierarchical clustering

Parameters: linkage type and the distance

K-means

Parameters: K and the distance

DBSCAN

Parameters: minPoints and epsilon
Hierarchical clustering

Using a **dendrogram** can extract **any number** of clusters.

Parameters: **linkage type** and the **distance**

**DBSCAN**

Parameters: **minPoints** and **epsilon**

**K-means**

Parameters: **K** and the **distance**

Parameters: centroids

Noise
Hierarchical clustering

Parameters: linkage type and the distance

Using a **dendrogram** can extract **any** number of clusters

Hierarchical clustering is **very slow**

**K-means**

Parameters: **K** and the distance

**DBSCAN**

Parameters: **minPoints** and **epsilon**

Overfitting
Hierarchical clustering

Hierarchical clustering is very slow \((N^3)\)

Parameters: linkage type and the distance

Using a dendrogram can extract any number of clusters

K-means

Parameters: \(K\) and the distance

Hierarchical clustering is very slow \((N^3)\)

DBSCAN

Parameters: \texttt{minPoints} and \texttt{epsilon}
Hierarchical clustering

- Parameters: minPoints and epsilon

Hierarchical clustering is very slow ($N^3$)

Using a dendrogram can extract any number of clusters

K-means

- Parameters: K and the distance

Hierarchical clustering is very slow ($N^3$)

DBSCAN

- Parameters: minPoints and epsilon

Noise
Hierarchical clustering

Using a **dendrogram** can extract **any number** of clusters.

Parameters: **linkage type** and the **distance**

Hierarchical clustering is **very slow** ($N^3$)

K-means

You will have **exactly K clusters**

Parameters: **K** and the **distance**

DBSCAN

Parameters: **minPoints** and **epsilon**
Hierarchical clustering is very slow ($N^3$).

Using a dendrogram can extract any number of clusters.

Parameters: linkage type and the distance.

K-means

You need to know $K$.

You will have exactly $K$ clusters.

Parameters: $K$ and the distance.

DBSCAN

Parameters: minPoints and epsilon.

noise
Hierarchical clustering

Hierarchical clustering is very slow ($N^3$)

Using a dendrogram can extract any number of clusters

Parameters: linkage type and the distance

K-means

You will have exactly $K$ clusters

Although NP-hard, it works surprisingly fast

Parameters: $K$ and the distance

DBSCAN

Parameters: $minPoints$ and epsilon

noise

centroids

Hierarchical clustering

- Parameters: linkage type and the distance

- Using a dendrogram can extract any number of clusters

K-means

- You need to know K

- You will have exactly K clusters

- Although NP-hard, it works surprisingly fast

DBSCAN

- Parameters are tedious to pick up

- Does not depend on K

- Parameters: minPoints and epsilon

Hierarchical clustering is very slow ($N^3$)
References

• Machine Learning by **Andrew Ng** ([https://www.coursera.org/learn/machine-learning](https://www.coursera.org/learn/machine-learning))

• Curse of dimensionality: [http://cleverowl.uk/2016/02/06/curse-of-dimensionality-explained/](http://cleverowl.uk/2016/02/06/curse-of-dimensionality-explained/)


• Stanford CS class: Convolutional Neural Networks for Visual Recognition by **Andrej Karpathy** ([http://cs231n.github.io/](http://cs231n.github.io/))

• Data Mining Course by **Jaak Vilo** at University of Tartu ([https://courses.cs.ut.ee/MTAT.03.183/2017_spring/uploads/Main/DM_05_Clustering.pdf](https://courses.cs.ut.ee/MTAT.03.183/2017_spring/uploads/Main/DM_05_Clustering.pdf))
What time is it now?
Kahoot!
That's all Folks!