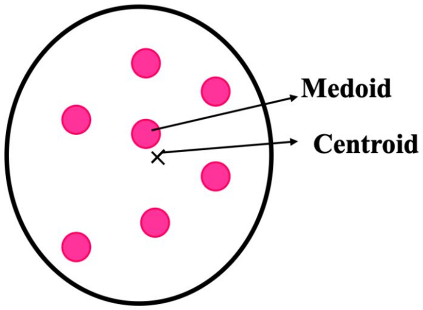
Unsupervised Learning (Clustering)

Supervised vs Unsupervised Learning

- **Supervised vs unsupervised learning**
  - **Supervised: We know the class labels and the number of classes. We want to build a classifier that can be used to predict the class of new (unlabelled) examples.**
  - **Unsupervised: We do not know the class labels and may not know the number of classes. We want to group similar examples together.**

Centroid and Medoid of a Cluster

- **Consider a cluster K of N points $\{p_1,..,p_N\}$**
- ***Centroid (means)*** **– the "middle" of the cluster** $C = \frac{\sum_{i=1}^{N} p_i}{N}$
  - **No need to be an actual data point in the cluster**
- ***Medoid M*** **– the centrally located data point in the cluster**



<u>Exercise 1.</u> K-means clustering (Homework)

Given is the one-dimensional dataset: {3, 5, 7, 10, 13}. Run the k-means clustering algorithm for 1 epoch to cluster this dataset into 2 clusters: K1 and K2. Assume that the initial seeds (cluster centers) are c1=3 and c2=10 and that the distance measure is the absolute distance between the examples.

K1 =
K2 =

Epoch 1 - start

3: seed of cluster K1

Exercise 1. ***K-means clustering (Homework)***

Given is the one-dimensional dataset: {3, 5, 7, 10, 13}. Run the k-means clustering algorithm for 1 epoch to cluster this dataset into 2 clusters: K1 and K2. Assume that the initial seeds (cluster centers) are c1=3 and c2=10 and that the distance measure is the absolute distance between the examples.

K1 = {3}
K2 =

Epoch 1 - start

3: seed of cluster K1 => 3 ∈ K1
5: d(c1=3, 5) = |5 - 3| = 2, d(c2=10, 5) = |10 - 5| = 5

***Exercise 1.* K-means clustering (Homework)**

Given is the one-dimensional dataset: {3, 5, 7, 10, 13}. Run the k-means clustering algorithm for 1 epoch to cluster this dataset into 2 clusters: K1 and K2. Assume that the initial seeds (cluster centers) are c1=3 and c2=10 and that the distance measure is the absolute distance between the examples.

K1 = {3}
K2 =

Epoch 1 - start

3: seed of cluster K1 => 3 ∈ K1
5: **d(c1=3, 5) = |5 - 3| = 2,** d(c2=10, 5) = |10 - 5| = 5

Smaller distance

Exercise 1. ***K-means clustering (Homework)***

Given is the one-dimensional dataset: {3, 5, 7, 10, 13}. Run the k-means clustering algorithm for 1 epoch to cluster this dataset into 2 clusters: K1 and K2. Assume that the initial seeds (cluster centers) are c1=3 and c2=10 and that the distance measure is the absolute distance between the examples.

K1 = {3, 5}
K2 =

Epoch 1 - start

3: seed of cluster K1 => 3 ∈ K1

5: **d(c1=3, 5) = |5 - 3| = 2,** d(c2=10, 5) = |10 - 5| = 5 => 5 ∈ K1

Smaller distance

***Exercise 1.* K-means clustering (Homework)**

Given is the one-dimensional dataset: {3, 5, 7, 10, 13}. Run the k-means clustering algorithm for 1 epoch to cluster this dataset into 2 clusters: K1 and K2. Assume that the initial seeds (cluster centers) are c1=3 and c2=10 and that the distance measure is the absolute distance between the examples.

K1 = {3, 5}
K2 = {7}

Epoch 1 - start

3: seed of cluster K1 => 3 ∈ K1
5: **d(c1=3, 5) = |5 - 3| = 2,** d(c2=10, 5) = |10 - 5| = 5 => 5 ∈ K1
7: d(c1=3, 7) = 4, **d(c2=10, 7) = 3** => 7 ∈ K2

<u>Exercise 1.</u> K-means clustering (Homework)

Given is the one-dimensional dataset: {3, 5, 7, 10, 13}. Run the k-means clustering algorithm for 1 epoch to cluster this dataset into 2 clusters: K1 and K2. Assume that the initial seeds (cluster centers) are c1=3 and c2=10 and that the distance measure is the absolute distance between the examples.

K1 = {3, 5}
K2 = {7, 10}

Epoch 1 - start

3: seed of cluster K1 => 3 ∈ K1
5: **d(c1=3, 5) = |5 - 3| = 2,** d(c2=10, 5) = |10 - 5| = 5 => 5 ∈ K1
7: d(c1=3, 7) = 4, **d(c2=10, 7) = 3** => 7 ∈ K2
10: seed of cluster K2 => 10 ∈ K2

Exercise 1. ***K-means clustering (Homework)***

Given is the one-dimensional dataset: {3, 5, 7, 10, 13}. Run the k-means clustering algorithm for 1 epoch to cluster this dataset into 2 clusters: K1 and K2. Assume that the initial seeds (cluster centers) are c1=3 and c2=10 and that the distance measure is the absolute distance between the examples.

K1 = {3, 5}
K2 = {7, 10, 13}

Epoch 1 - start

3: seed of cluster K1 => 3 ∈ K1
5: **d(c1=3, 5) = |5 - 3| = 2,** d(c2=10, 5) = |10 - 5| = 5 => 5 ∈ K1
7: d(c1=3, 7) = 4, **d(c2=10, 7) = 3** => 7 ∈ K2
10: seed of cluster K2 => 10 ∈ K2
13: d(c1=3, 13) = 10, **d(c2=10, 13) = 3** => 13 ∈ K2

Exercise 1. K-means clustering (Homework)

Given is the one-dimensional dataset: {3, 5, 7, 10, 13}. Run the k-means clustering algorithm for 1 epoch to cluster this dataset into 2 clusters: K1 and K2. Assume that the initial seeds (cluster centers) are c1=3 and c2=10 and that the distance measure is the absolute distance between the examples.

K1 = {3, 5} **K2 = {7, 10, 13}**	**c1_new = 4**

Epoch 1 - start

3: seed of cluster K1 => 3 ∈ K1
5: **d(c1=3, 5) = |5 - 3| = 2,** d(c2=10, 5) = |10 - 5| = 5 => 5 ∈ K1
7: d(c1=3, 7) = 4, **d(c2=10, 7) = 3** => 7 ∈ K2
10: seed of cluster K2 => 10 ∈ K2
13: d(c1=3, 13) = 10, **d(c2=10, 13) = 3** => 13 ∈ K2

New centroids:

Exercise 1. ***K-means clustering (Homework)***

Given is the one-dimensional dataset: {3, 5, 7, 10, 13}. Run the k-means clustering algorithm for 1 epoch to cluster this dataset into 2 clusters: K1 and K2. Assume that the initial seeds (cluster centers) are c1=3 and c2=10 and that the distance measure is the absolute distance between the examples.

K1 = {3, 5}	**c1_new = 4**
K2 = {7, 10, 13}	

Epoch 1 - start

3: seed of cluster K1 => 3 ∈ K1
5: **d(c1=3, 5) = |5 - 3| = 2,** d(c2=10, 5) = |10 - 5| = 5 => 5 ∈ K1
7: d(c1=3, 7) = 4, **d(c2=10, 7) = 3** => 7 ∈ K2
10: seed of cluster K2 => 10 ∈ K2
13: d(c1=3, 13) = 10, **d(c2=10, 13) = 3** => 13 ∈ K2

New centroids:

- c1_new = (3+5)/2 = 4

<u>Exercise 1.</u> K-means clustering (Homework)

Given is the one-dimensional dataset: {3, 5, 7, 10, 13}. Run the k-means clustering algorithm for 1 epoch to cluster this dataset into 2 clusters: K1 and K2. Assume that the initial seeds (cluster centers) are c1=3 and c2=10 and that the distance measure is the absolute distance between the examples.

K1 = {3, 5}	**c1_new = 4**
K2 = {7, 10, 13}	**c2_new = 10**

Epoch 1 - start

3: seed of cluster K1 => 3 ∈ K1
5: **d(c1=3, 5) = |5 - 3| = 2,** d(c2=10, 5) = |10 - 5| = 5 => 5 ∈ K1
7: d(c1=3, 7) = 4, **d(c2=10, 7) = 3** => 7 ∈ K2
10: seed of cluster K2 => 10 ∈ K2
13: d(c1=3, 13) = 10, **d(c2=10, 13) = 3** => 13 ∈ K2

New centroids:

- c1_new = (3+5)/2 = 4
- c2_new = (7+10+13)/3 = 10

Nearest Neighbour Clustering

*not to be confused with K nearest-neighbour - the supervised ML algorithm

- **Idea: Items are iteratively merged into clusters**
  - **The first item forms a cluster of itself**
  - **A new item is either merged with an existing cluster or forms a new cluster of itself depending on how close it is to the existing clusters**

    if distance(new_item, existing_cluster) $\leq$ threshold t -> merge

    else new_item forms a cluster of itself

- Advantage compared to k-means: don’t need to specify the number of clusters

Exercise 2. ***Nearest neighbor clustering***

Use the Nearest Neighbor clustering algorithm to cluster examples A, B, C and D described by the following distance matrix. Suppose that the threshold *t* is 3.

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	4
D				0

- **The first item forms a cluster of itself**
- **A new item is either merged with an existing cluster or forms a new cluster of itself depending on how close it is to the existing clusters**

  if distance(new_item, existing_cluster) ≤ threshold t -> merge

  else new_item forms a cluster of itself

Clusters:

Explanation:

Exercise 2. ***Nearest neighbor clustering***

Use the Nearest Neighbor clustering algorithm to cluster examples A, B, C and D described by the following distance matrix. Suppose that the threshold t is 3.

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	4
D				0

- **The first item forms a cluster of itself**
- **A new item is either merged with an existing cluster or forms a new cluster of itself depending on how close it is to the existing clusters**

  if distance(new_item, existing_cluster) ≤ threshold t -> merge

  else new_item forms a cluster of itself

Clusters:

K1 = {A}

Explanation:

- A is the first item => A forms a cluster of itself

Exercise 2. ***Nearest neighbor clustering***

Use the Nearest Neighbor clustering algorithm to cluster examples A, B, C and D described by the following distance matrix. Suppose that the threshold *t* is 3.

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	4
D				0

- **The first item forms a cluster of itself**
- **A new item is either merged with an existing cluster or forms a new cluster of itself depending on how close it is to the existing clusters**

  if distance(new_item, existing_cluster) ≤ threshold t -> merge

  else new_item forms a cluster of itself

Clusters:

K1 = {A}

Explanation:

Check if **B** should be added to K1 or be placed in a new cluster?

***Exercise 2.* Nearest neighbor clustering**

Use the Nearest Neighbor clustering algorithm to cluster examples A, B, C and D described by the following distance matrix. Suppose that the threshold t is 3.

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	4
D				0

- **The first item forms a cluster of itself**
- **A new item is either merged with an existing cluster or forms a new cluster of itself depending on how close it is to the existing clusters**

  if distance(new_item, existing_cluster) ≤ threshold t -> merge

  else new_item forms a cluster of itself

Clusters:

K1 = {A}

Explanation:

Check if **B** should be added to K1 or be placed in a new cluster?

d(A,B) = 1

Exercise 2. *Nearest neighbor clustering*

Use the Nearest Neighbor clustering algorithm to cluster examples A, B, C and D described by the following distance matrix. Suppose that the threshold *t* is 3.

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	4
D				0

- **The first item forms a cluster of itself**
- **A new item is either merged with an existing cluster or forms a new cluster of itself depending on how close it is to the existing clusters**

  if distance(new_item, existing_cluster) ≤ threshold t -> merge

  else new_item forms a cluster of itself

Clusters:

K1 = {A}

Explanation:

Check if **B** should be added to K1 or be placed in a new cluster?

$d(A,B) = 1 < t \quad (t = 3)$

Exercise 2. Nearest neighbor clustering

Use the Nearest Neighbor clustering algorithm to cluster examples A, B, C and D described by the following distance matrix. Suppose that the threshold *t* is 3.

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	4
D				0

- **The first item forms a cluster of itself**
- **A new item is either merged with an existing cluster or forms a new cluster of itself depending on how close it is to the existing clusters**

  if distance(new_item, existing_cluster) ≤ threshold t -> merge

  else new_item forms a cluster of itself

Clusters:

K1 = {A, B}

Explanation:

Check if **B** should be added to K1 or be placed in a new cluster?

$d(A,B) = 1 < t$

=> B is added to K1

Exercise 2. *Nearest neighbor clustering*

Use the Nearest Neighbor clustering algorithm to cluster examples A, B, C and D described by the following distance matrix. Suppose that the threshold t is 3.

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	4
D				0

- **The first item forms a cluster of itself**
- **A new item is either merged with an existing cluster or forms a new cluster of itself depending on how close it is to the existing clusters**

  if distance(new_item, existing_cluster) ≤ threshold t -> merge

  else new_item forms a cluster of itself

Clusters:

K1 = {A, B}

Explanation:

Check if **B** should be added to K1 or be placed in a new cluster?

d(A,B) = 1 < t => B is added to K1

Next: **C** - added to K1 or placed in a new cluster?

Exercise 2. Nearest neighbor clustering

Use the Nearest Neighbor clustering algorithm to cluster examples A, B, C and D described by the following distance matrix. Suppose that the threshold *t* is 3.

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	4
D				0

- **The first item forms a cluster of itself**
- **A new item is either merged with an existing cluster or forms a new cluster of itself depending on how close it is to the existing clusters**

  if distance(new_item, existing_cluster) ≤ threshold t -> merge

  else new_item forms a cluster of itself

Clusters:

K1 = {A, B}

Explanation:

Check if **B** should be added to K1 or be placed in a new cluster?

d(A,B) = 1 < t => B is added to K1

C - added to K1 or placed in a new cluster?

d(C,A)=4, d(C,B)=2

Exercise 2. ***Nearest neighbor clustering***

Use the Nearest Neighbor clustering algorithm to cluster examples A, B, C and D described by the following distance matrix. Suppose that the threshold t is 3.

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	4
D				0

- **The first item forms a cluster of itself**
- **A new item is either merged with an existing cluster or forms a new cluster of itself depending on how close it is to the existing clusters**

  if distance(new_item, existing_cluster) ≤ threshold t -> merge

  else new_item forms a cluster of itself

Clusters:

K1 = {A, B}

Explanation:

Check if **B** should be added to K1 or be placed in a new cluster?

d(A,B) = 1 < t => B is added to K1

C - added to K1 or placed in a new cluster?

d(C,A)=4, **d(C,B)=2** → **the smallest, also smaller than t**

Exercise 2. ***Nearest neighbor clustering***

Use the Nearest Neighbor clustering algorithm to cluster examples A, B, C and D described by the following distance matrix. Suppose that the threshold t is 3.

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	4
D				0

- **The first item forms a cluster of itself**
- **A new item is either merged with an existing cluster or forms a new cluster of itself depending on how close it is to the existing clusters**

  if distance(new_item, existing_cluster) ≤ threshold t -> merge

  else new_item forms a cluster of itself

Clusters:

K1 = {A, B, C}

Explanation:

Check if **B** should be added to K1 or be placed in a new cluster?

d(A,B) = 1 < t => B is added to K1

C - added to K1 or placed in a new cluster?

d(C,A)=4, **d(C,B)=2** **the smallest, also smaller than t**

=> C is added to K1

Exercise 2. Nearest neighbor clustering

Use the Nearest Neighbor clustering algorithm to cluster examples A, B, C and D described by the following distance matrix. Suppose that the threshold t is 3.

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	4
D				0

- **The first item forms a cluster of itself**
- **A new item is either merged with an existing cluster or forms a new cluster of itself depending on how close it is to the existing clusters**

  if distance(new_item, existing_cluster) ≤ threshold t -> merge

  else new_item forms a cluster of itself

Clusters:

K1 = {A, B, C}

Explanation:

Check if **B** should be added to K1 or be placed in a new cluster?

d(A,B) = 1 < t => B is added to K1

C - added to K1 or placed in a new cluster?

d(C,A)=4, **d(C,B)=2 => C is added to K1**

Next: **D** - added to K1 or placed in a new cluster?

Exercise 2. Nearest neighbor clustering

Use the Nearest Neighbor clustering algorithm to cluster examples A, B, C and D described by the following distance matrix. Suppose that the threshold t is 3.

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	4
D				0

- **The first item forms a cluster of itself**
- **A new item is either merged with an existing cluster or forms a new cluster of itself depending on how close it is to the existing clusters**

  if distance(new_item, existing_cluster) ≤ threshold t -> merge

  else new_item forms a cluster of itself

Clusters:

K1 = {A, B, C}

K2 = {D}

Explanation:

Check if **B** should be added to K1 or be placed in a new cluster?

d(A,B) = 1 < t => B is added to K1

C - added to K1 or placed in a new cluster?

d(C,A)=4, **d(C,B)=2** **=> C is added to K1**

D - added to K1 or placed in a new cluster?

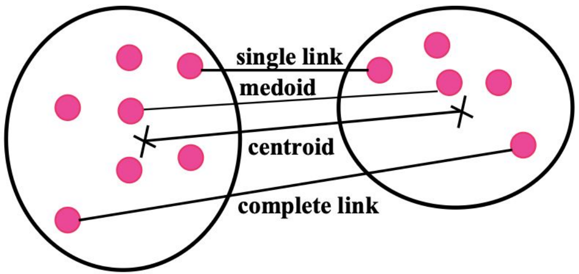
d(D,A)=5, d(D,B)=6, d(D,C)=4

=> Smallest d is: **d(D,C) = 4 > t**

=> D is placed in a new cluster K2

Agglomerative Clustering

- A Hierarchical Clustering Algorithm
- Agglomerative (bottom-up) - **merges** clusters iteratively
  - Start with each item in its own cluster
  - Then, iteratively merge clusters until all items belong to one cluster
- Advantage compared to k-means: don’t need to specify the number of clusters



***Exercise 3.* Hierarchical clustering – single link agglomerative algorithm**

Use the **single link** agglomerative clustering to group the data described by the following distance matrix. Draw the dendrogram.

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	3
D				0

Dendrogram:

Solution:

Level 0: each item is in its own cluster

{A}, {B}, {C}, {D}

Exercise 3. ***Hierarchical clustering – single link agglomerative algorithm***

Use the **single link** agglomerative clustering to group the data described by the following distance matrix. Draw the dendrogram.

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	3
D				0

Dendrogram:

Solution:

Level 0: each item is in its own cluster

{A}, {B}, {C}, {D}

Level **1**: merge A and B as d(A,B) ≤ **1**

{A, B}, {C}, {D}

<u>Exercise 3.</u> Hierarchical clustering – single link agglomerative algorithm

Use the **single link** agglomerative clustering to group the data described by the following distance matrix. Draw the dendrogram.

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	3
D				0

	AB	C	D
AB	0	?	?
C		0	?
D			0

<u>Dendrogram:</u>

<u>Solution:</u>

Level 0: each item is in its own cluster

{A}, {B}, {C}, {D}

Level **1**: merge A and B as d(A,B) ≤ **1**

{A, B}, {C}, {D}

Next we need to update the distance matrix

Exercise 3. ***Hierarchical clustering – single link agglomerative algorithm***

Use the **single link** agglomerative clustering to group the data described by the following distance matrix. Draw the dendrogram.

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	3
D				0

→

	AB	C	D
AB	0	?	?
C		0	?
D			0

Dendrogram:

Solution:

Level 0: each item is in its own cluster

{A}, {B}, {C}, {D}

Level **1**: merge A and B as d(A,B) < **1**

{A, B}, {C}, {D}

Next we need to update the distance matrix

- A and B is now in one cluster

Exercise 3. ***Hierarchical clustering – single link agglomerative algorithm***

Use the **single link** agglomerative clustering to group the data described by the following distance matrix. Draw the dendrogram.

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	3
D				0

	AB	C	D
AB	0	?	?
C		0	3
D			0

Dendrogram:

Solution:

Level 0: each item is in its own cluster

{A}, {B}, {C}, {D}

Level **1**: merge A and B as d(A,B) < **1**

{A, B}, {C}, {D}

Next we need to update the distance matrix

- A and B is now in one cluster
- d(C,D) is still the same

<u>Exercise 3.</u> Hierarchical clustering – single link agglomerative algorithm

Use the **single link** agglomerative clustering to group the data described by the following distance matrix. Draw the dendrogram.

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	3
D				0

	AB	C	D
AB	0	2	?
C		0	3
D			0

<u>Dendrogram:</u>

<u>Solution:</u>

Level 0: each item is in its own cluster

{A}, {B}, {C}, {D}

Level **1**: merge A and B as d(A,B) < **1**

{A, B}, {C}, {D}

Next we need to update the distance matrix

- A and B is now in one cluster
- d(C,D) is still the same
- d({A,B}, C) using <u>single link</u> is **min**(d(A,C), d(B,C))=**min**(4,2)=**2**

Exercise 3. *Hierarchical clustering – single link agglomerative algorithm*

Use the **single link** agglomerative clustering to group the data described by the following distance matrix. Draw the dendrogram.

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	3
D				0

→

	AB	C	D
AB	0	2	5
C		0	3
D			0

Dendrogram:

Solution:

Level 0: each item is in its own cluster

{A}, {B}, {C}, {D}

Level **1**: merge A and B as d(A,B) < **1**

{A, B}, {C}, {D}

Next we need to update the distance matrix

- A and B is now in one cluster
- d(C,D) is still the same
- d({A,B}, C) using single link is **min**(d(A,C), d(B,C))=**min**(4,2)=**2**
- d({A,B}, D) using single link is **min**(d(A,D), d(B,D))=**min**(5,6)=**5**

Your turn to continue from level 2 until all items are in one cluster!

See solution

Exercise 4. *Hierarchical clustering – complete link agglomerative algorithm*

The same task as in the previous exercise but using the **complete link** distance measure.

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	3
D				0

Complete link (MAX) – the largest pairwise distance between elements from each cluster

Exercise 4. *Hierarchical clustering – complete link agglomerative algorithm*

The same task as in the previous exercise but using the **complete link** distance measure.

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	3
D				0

Complete link (MAX) – the largest pairwise distance between elements from each cluster

The updated matrix is:

	AB	C	D
AB	0	4	6
C		0	3
D			0

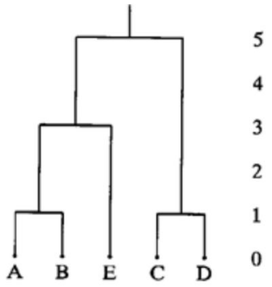
See solution

Single Link vs. Complete Link Algorithm

- **Single link (MIN) suffers from the so called *chain effect***
  - **2 clusters are merged if <u>only 2</u> of their elements are close to each other**
  - **There may be elements in the 2 clusters that are far from each other but this has no effect on the algorithm**
  - **Thus, the clusters may contain points that are not related to each other but simply happen to be near points that are close to each other**

  - **More sensitive to noise and outliers**

Dendrogram for our example



- **Complete link (MAX) – the distance between 2 clusters is the <u>largest</u> distance between an element in one cluster and an element in another**
  - **Generates more compact clusters**
  - **Less sensitive to noise and outliers**