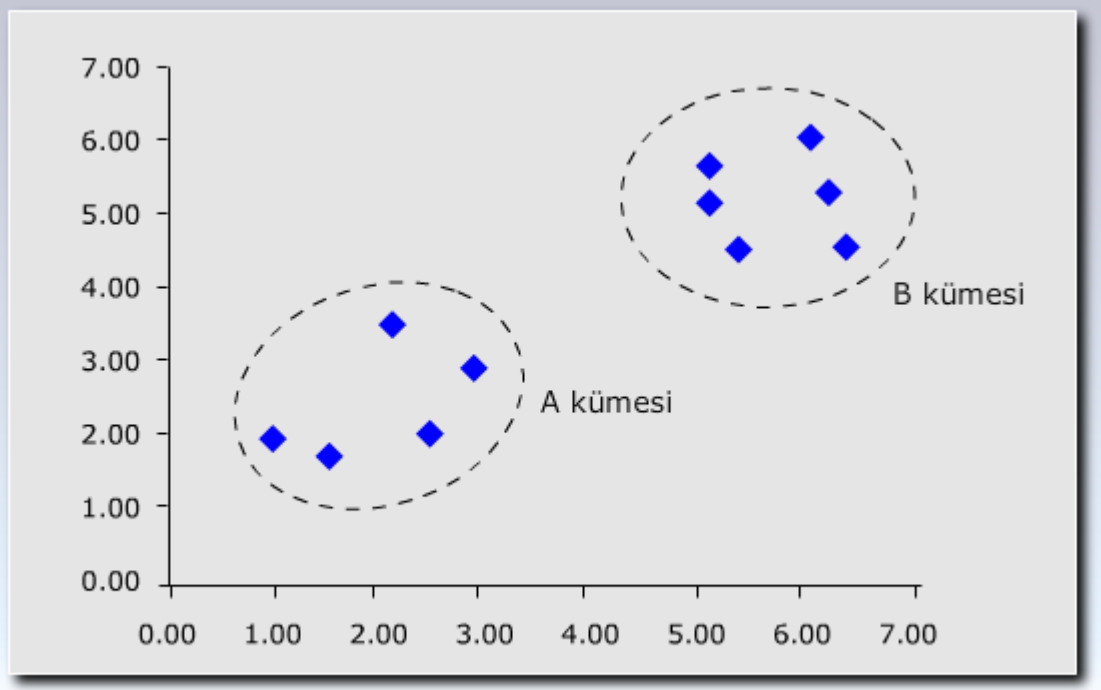
Kümeleme

Kümeleme

Verilerin birbirine benzeyen kısımlarının gruplandırılmasına kümeleme adını veriyoruz. Kümeleme çözümlemeleri veri madenciliğinde geniş bir uygulama alanı bulmuştur. Bu ders kapsamında verilerin birbirine olan uzaklıklarını esas alarak **hiyerarşik** ve **hiyerarşik olmayan** kümeleme biçiminde geliştirilmiş iki algoritma türü anlatılmaktadır.

Kümeleme Çözümlemesi

Veri madenciliğinin önemli konuları arasında yer alan **kümeleme çözümlemesi** (*cluster analysis*), verileri birbirleriyle benzer alt kümelere ayırma işlemi olarak bilinmektedir. Uygulamada çok sayıda kümeleme yöntemi kullanılmaktadır. Bu yöntemler, değişkenler arasındaki benzerliklerden yada farklılıklardan yararlanarak bir kümeyi alt kümelere ayırmakta kullanılmaktadır.



Şekil 6.1: A ve B gibi iki kümenin görünümü.

Kümeleme

- Kümeleme çözümlemeleri istatistikte başvurulan yöntemlerdir. Aslında kümeleme çözümlemeleri birbirine benzeyen gözlem değerlerinin ayrılarak sınıflandırılmasını sağlayan **çok değişkenli çözümleme yöntemleri** olarak karşımıza çıkmaktadır.
- Kümeleme çözümlemeleri **pazarlama faaliyetlerinde** sıkça kullanılır. Örneğin bir mamulden farklı beklentilerine göre müşterileri kümelere ayrılabilir. Bunun dışında belirli ürünleri kullanıcıların davranış biçimine göre gruplandırmak söz konusu olabilir. Böylece çözümleme ile elde edilen sonuçlara bakılarak pazarlama stratejisi belirlenebilir.

Kümeleme

Uzaklık Ölçüleri

Kümeleme yöntemlerinin birçoğu, gözlem değerleri arasındaki uzaklıkların hesaplanması esasına dayanmaktadır. O nedenle iki nokta arasındaki uzaklığı hesaplayan bağıntılara gereksinim vardır. Çeşitli değişkenlerden oluşan gözlem değerlerini bir **X** matrisi biçiminde gösterebiliriz.
Örneğin üç değişken ve 5 gözlem değerinden oluşan matris şu şekilde ifade edilebilir

$$X=\begin{bmatrix} \chi_{11} & \chi_{12} & \chi_{13} \\ \chi_{21} & \chi_{22} & \chi_{23} \\ \chi_{31} & \chi_{32} & \chi_{33} \\ \chi_{41} & \chi_{42} & \chi_{43} \\ \chi_{51} & \chi_{52} & \chi_{53} \end{bmatrix}$$

Burada birinci gözlem noktasının konumu (x_{11},x_{12},x_{13}) biçimindedir. İkinci gözlemin konumu ise (x_{21},x_{22},x_{23}) olarak ifade edilebilir. Bu iki nokta arasındaki uzaklık ise d(**1,2**) biçiminde yazılabilir. Yukarıdaki X matrisinin her bir satırının diğerine olan uzaklığı d(i,j) biçiminde ifade edilecek olursa, simetrik D uzaklıklar matrisi şu şekilde yazılabilir

Kümeleme

Aşağıdaki matrisin üst kısmı alt kısmının simetriği olduğundan ayrıca yazılmamıştır. Bu durumda **d(i,j)** = **d(j,i)** olduğu bilinir. Kümeleme çözümlemelerinde birçok uzaklık bağıntısı kullanılabilmektedir. Bunlardan üç tanesine aşağıda yer veriyoruz.

$$D = \begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ d(4,1) & d(4,2) & d(4,3) & 0 & \\ d(5,1) & d(5,2) & d(5,3) & d(5,4) & 0 \end{bmatrix}$$

Simetrik

Kümeleme

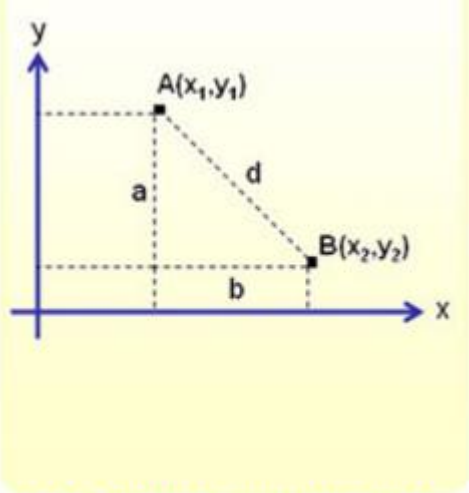
Öklid Uzaklığı

- Uygulamada en çok kullanılan uzaklık ölçüsü **Öklid uzaklık bağıntısı** adıyla bilinmektedir. Bu uzaklık, yandaki şekil üzerinde görüldüğü gibi, iki boyutlu uzayda Pisagor teoreminin bir uygulaması olarak karşımıza çıkmaktadır.
- A ve B noktaları arasındaki Öklid uzaklığı şu şekilde olacaktır

$$d(A,B) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Bu bağıntı genelleştirilecek olursak, i ve j noktaları için şu şekilde bir bağıntıya ulaşılır:

$$d(i,j) = \sqrt{\sum_{k=1}^{p} \left(x_{ik} - x_{jk}\right)^2}$$



Şekil 6.2: İki nokta arasındaki d uzaklığı.

Kümeleme

Manhattan Uzaklığı

Diğer bir uzaklık ölçüsü Manhattan uzaklığıdır. Bu uzaklık, gözlemler arasındaki mutlak uzaklıkların toplamı alınarak hesaplanır. Söz konusu uzaklık şu şekilde ifade edilir

$$d(i,j)=\sum_{k=1}^{p}\left(\left|x_{ik}-x_{jk}\right|\right) \qquad \text{i,j=1,2,...,n; k=1,2,...,p}$$

Kümeleme

Minkowski Uzaklığı

p sayıda değişken göz önüne alınarak gözlem değerleri arasındaki uzaklığın hesaplanması söz konusu ise **Minkowski uzaklık bağıntısı** kullanılabilir. Söz konusu uzaklık şu şekilde hesaplanır:

$$d(i,j)=\left[\sum_{k=1}^{p}\left(\left|x_{ik}-x_{jk}\right|^{m}\right)\right]^{\frac{1}{m}} \quad i,j=1,2,\ldots,n;\ k=1,2,\ldots,p$$

Burada **m=2** yazılarak Öklid uzaklık bağıntısı elde edilebilir.

Kümeleme

Örnek

A,**B** ve **C** gibi üç değişkenden oluşan aşağıdaki gözlemleri göz önüne alalım. Bu gözlem noktalarının her birinin birbirine olan uzaklığını farklı uzaklık ölçüleriyle elde etmek istiyoruz.

Gözlem	A	B	C
1	2	3	1
2	4	1	3
3	5	7	3
4	4	8	2
5	3	9	5

Tablo 6.1: Gözlem değerleri.

Örneğimizin çözümünü üç uzaklık bağıntısını da kullanarak yapalım.

Kümeleme

Öklid Uzaklığı

Burada yer alan üç değişken için, i ve j gözlem noktaları ve p=3 olmak üzere Öklid uzaklık bağıntısını şu şekilde tanımlayabiliriz:

$$d(i,j)=\sqrt{(x_{i1}-x_{j1})^2+(x_{i2}-x_{j2})^2+(x_{i3}-x_{j3})^2}$$

İkinci gözlem ile birinci gözlem yani arasındaki uzaklık şu şekilde hesaplanır:

$$d(2,1)=\sqrt{(x_{21}-x_{11})^2+(x_{22}-x_{12})^2+(x_{23}-x_{13})^2}$$
$$=\sqrt{(4-2)^2+(1-3)^2+(3-1)^2}=3.46$$

Üçüncü gözlem ile birinci gözlem yani arasındaki uzaklık ise şu şekilde hesaplanır:

$$d(3,1)=\sqrt{(x_{31}-x_{11})^2+(x_{32}-x_{12})^2+(x_{33}-x_{13})^2}$$
$$=\sqrt{(5-2)^2+(7-3)^2+(3-1)^2}=5.39$$

Benzer biçimde her bir gözlem değeri arasındaki Öklid uzaklıkları hesaplanarak aşağıdaki sonuçlar elde edilir.

Gözlem	1	2	3	4	5
1	0,00				
2	3,46	0,00			
3	5,39	6,08	0,00		
4	5,48	7,07	1,73	0,00	
5	7,28	8,31	3,46	3,32	0,00

Tablo 6.2: Öklid uzaklıkları.

Kümeleme

Manhattan Uzaklığı

Söz konusu verileri kullanarak Manhattan uzaklığını hesaplayabiliriz. Üç değişken için Manhattan uzaklık bağıntısı şu biçimdedir:

$$d(i,j)=|x_{i1}-x_{j1}|+|x_{i2}-x_{j2}|+|x_{i3}-x_{j3}|$$

Bu bağıntı yardımıyla ikinci gözlem ile birinci gözlem arasındaki Manhattan uzaklığını elde edelim:

$$d(2,1)=|4-2|+|1-3|+|3-1|=6$$

Üçüncü gözlem ile birinci gözlem arasındaki Manhattan uzaklığı ise şu şekildedir:

$$d(3,1)=|5-2|+|7-3|+|3-1|=9$$

Benzer biçimde diğer tüm gözlemlerin birbirlerine olan uzaklıkları tek tek hesaplanırsa aşağıdaki sonuç elde edilir:

Gözlem	1	2	3	4	5
1	0,00				
2	6,00	0,00			
3	9,00	7,00	0,00		
4	8,00	8,00	3,00	0,00	
5	11,00	11,00	6,00	5,00	0,00

Tablo 6.3: Manhattan uzaklıkları.

Kümeleme

Minskowski Uzaklığı

Gözlem değerlerini yeniden ele alalım. Bu kez **Minkowski uzaklık bağıntısını** kullanarak tüm gözlemler arasındaki uzaklıkları elde edeceğiz. Üç değişken için Minkowski uzaklık bağıntısı şu şekli alacaktır:

$$d(i,j) = \left[|x_{i1} - x_{j1}|^m + |x_{i2} - x_{j2}|^m + |x_{i3} - x_{j3}|^m \right]^{1/m}$$

Bu bağıntıdan yararlanarak **m=3** varsayımı altında ikinci gözlem ile birinci gözlem arasındaki uzaklık şu şekilde hesaplanır:

$$d(2,1) = \left[|x_{21} - x_{11}|^3 + |x_{22} - x_{12}|^3 + |x_{23} - x_{13}|^3 \right]^{1/3}$$

$$= \left[|4-2|^3 + |1-3|^3 + |3-1|^3 \right]^{1/3} = 2.88$$

Benzer biçimde tüm gözlem noktaları arasındaki uzaklıklar hesaplanarak aşağıdaki sonuca ulaşılır:

Bu kez üçüncü gözlem ile birinci gözlem arasındaki Minkowski uzaklığını bulalım:

$$d(3,1) = \left[|x_{31} - x_{11}|^3 + |x_{32} - x_{12}|^3 + |x_{33} - x_{13}|^3 \right]^{1/3}$$

$$= \left[|5-2|^3 + |7-3|^3 + |3-1|^3 \right]^{1/3} = 4.63$$

Kümeleme

Minskowski Uzaklığı

Gözlem değerlerini yeniden ele alalım. Bu kez **Minkowski uzaklık bağıntısını** kullanarak tüm gözlemler arasındaki uzaklıkları elde edeceğiz. Üç değişken için Minkowski uzaklık bağıntısı şu şekli alacaktır:

Gözlem	1	2	3	4	5
1	0,00				
2	2,88	0,00			
3	4,63	6,01	0,00		
4	5,12	7,01	1,44	0,00	
5	6,55	8,05	2,88	3,07	0,00

Tablo 6.4: Minkowski uzaklıkları

Kümeleme

Hiyerarşik Kümeleme

- Birçok kümeleme yönteminden söz edilebilir. Hiyerarşik kümeleme ve hiyerarşik olmayan kümeleme yöntemlerini bu ders kapsamı içinde ele alarak inceleyeceğiz.
- Hiyerarşik kümeleme yöntemleri, kümelerin bir ana küme olarak ele alınması ve sonra aşamalı olarak içerdiği alt kümelere ayrılması veya ayrı ayrı ele alınan kümelerin aşamalı olarak bir küme biçiminde birleştirilmesi esasına dayanır.

Kümeleme

Birleştirici Hiyerarşik Yöntemler

Ayrı ayrı ele alınan kümelerin aşamalı olarak birleştirilmesini sağlayan yöntemlerdir. Bu grupta birçok hiyerarşik yöntem bulunmaktadır. Söz konusu yöntemlerden aşağıda belirtilenleri ele alarak inceleyeceğiz:

- *En yakın komşu algoritması*
- *En uzak komşu algoritması*

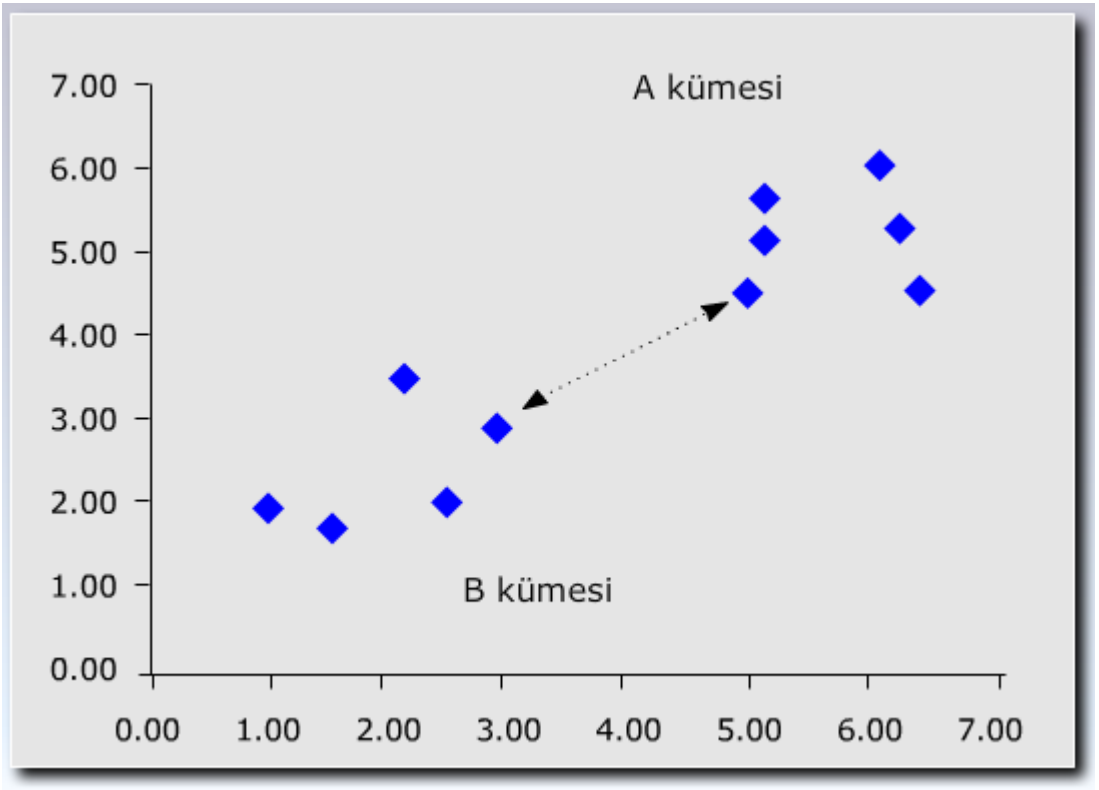
6.3.2. En yakın Komşu Algoritması

En yakın komşu yöntemine (*nearest neighbor method*) **"tek bağlantı kümeleme yöntemi"** adı da verilmektedir. Başlangıçta tüm gözlem değerleri birer küme olarak değerlendirilir. Adım adım bu kümeler birleştirilerek yeni kümeler elde edilir.
Bu yöntemde öncelikle gözlemler arasındaki uzaklıklar belirlenir. i ve j gözlemleri arasındaki uzaklıkların belirlenmesinde Öklid uzaklık bağıntısı kullanılabilir:

$$d(i,j) = \sqrt{\sum_{k=1}^{p} \left(x_{ij} - x_{jk}\right)^2}$$

Uzaklıklar göz önüne alınarak Min*d*(i,j) seçilir. Bu uzaklıkla ilgili satırlar birleştirilerek yeni bir küme elde edilir. Yeni duruma göre uzaklıkların yeniden hesaplanması gerekir. Tek bir gözlemden oluşan kümeler arasındaki uzaklıkları yukarıdaki formül ile doğrudan hesaplayabiliriz. Ancak birden fazla gözlem değerine sahip olan iki küme arasındaki uzaklığın belirlenmesi gerektiğinde farklı bir yol izlenir.
İki kümenin içerdiği gözlemler arasında **birbirine en yakın olanların uzaklığı** iki kümenin birbirine olan uzaklığı olarak kabul edilir.

Kümeleme



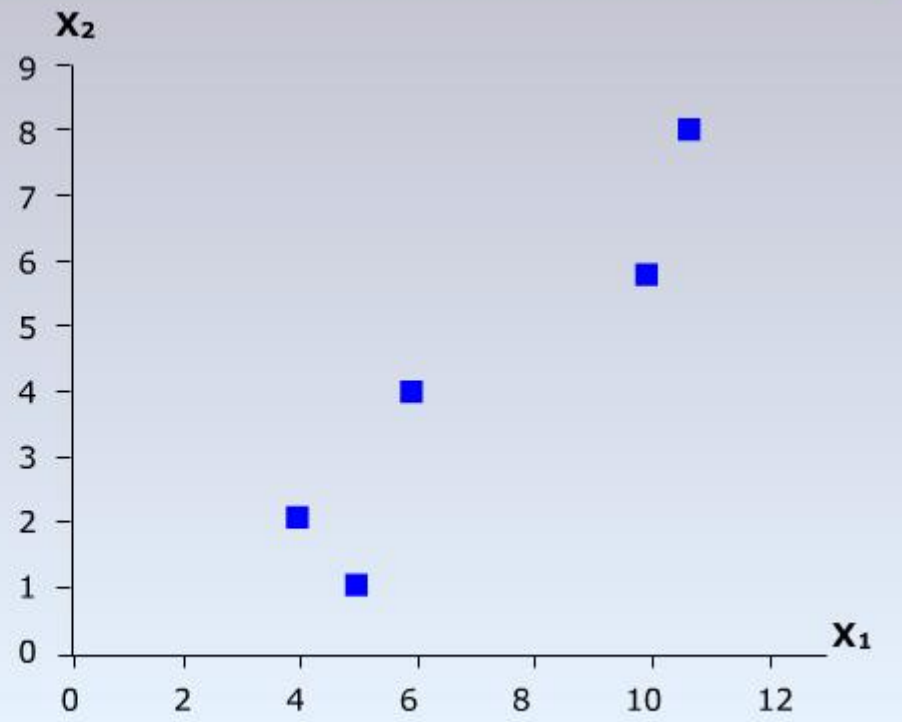
En yakın komşu algoritmasında iki kümenin birbirine en yakın gözlemleri arasındaki uzaklık iki kümenin birbirine olan uzaklığı olarak değerlendirilir.

6.3.2.1. Uygulama 1

Aşağıdaki tabloda verilen beş adet gözlemi göz önüne alalım. Bu veriler üzerinde **en yakın komşu algoritmasını kullanarak kümeleme işlemlerini yapmak istiyoruz**.

Gözlemler	X_1	X_2
1	4	2
2	6	4
3	5	1
4	10	6
5	11	8

Tablo 6.4: Gözlem değerleri



ŞEKİL-6.4: Gözlem değerlerinin grafik üzerindeki görünümü.

En yakın komşu algoritmasını adım adım uygulayalım.

Kümeleme

Adım 1

Öncelikle uzaklık tablosunun (matrisinin) hesaplanması gerekiyor. Uzaklık tablosu için çeşitli uzaklık ölçüleri kullanılabilir. Biz Öklid uzaklık ölçüsünü bu amaçla kullanmak istiyoruz. Söz konusu uzaklık bağıntısının k değişken sayısını göstermek üzere şu şekilde olduğunu biliyoruz.

Bu formül yardımıyla aşağıdaki hesaplamalar yapılır:

$$d(1,2) = \sqrt{(4-6)^2 + (2-4)^2} = 2.83$$
$$d(1,3) = \sqrt{(4-5)^2 + (2-1)^2} = 1.41$$
$$d(1,4) = \sqrt{(4-10)^2 + (2-6)^2} = 7.21$$
$$d(1,5) = \sqrt{(4-11)^2 + (2-8)^2} = 9.22$$
$$d(2,3) = \sqrt{(6-5)^2 + (4-1)^2} = 3.16$$
$$d(2,4) = \sqrt{(6-10)^2 + (4-6)^2} = 4.47$$
$$d(2,5) = \sqrt{(6-11)^2 + (4-8)^2} = 7.20$$
$$d(3,4) = \sqrt{(5-10)^2 + (1-6)^2} = 7.07$$
$$d(3,5) = \sqrt{(5-11)^2 + (1-8)^2} = 9.22$$
$$d(4,5) = \sqrt{(10-11)^2 + (6-8)^2} = 2.24$$

Kümeleme

Bu durumda gözlemlere ilişkin uzaklıklar matrisi şu şekilde olacaktır:

Gözlemler	1	2	3	4	5
1					
2	2.83				
3	1.41	3.16			
4	7.21	4.47	7.07		
5	9.22	7.20	9.22	2.24	

Tablo 6.5: Uzaklıklar tablosu

Adım 2

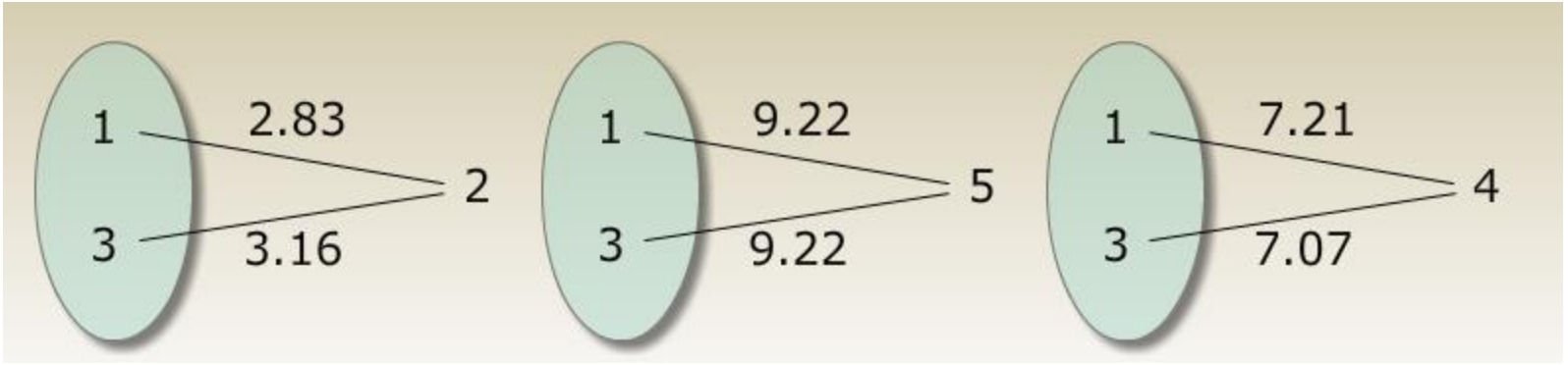
Uzaklıklar tablosunda Min*d*(i,j) hücresinin belirlenmesi gerekiyor. Tablo 6.5 incelendiğinde Mind(i,j)=1.41 olduğu görülür. O halde bu değerin ilgili olduğu 1 ve 3 numaralı gözlemeler ele alınır. Bu iki değer birleştirilerek (**1,3**) **kümesi** elde edilir.

Şimdi bu yeni elde edilen kümeye göre uzaklıklar matrisini yeniden gözden geçirmemiz gerekmektedir. Çünkü (**1,3**) kümesi ile diğer gözlemler arasındaki uzaklıkları belirlememiz söz konusudur. Bunun için, söz konusu kümenin elemanları ile diğer gözlemler eşlenerek içlerinden en küçük olanlar, yani birbirine en yakın olan gözlemlerle ilgili olan uzaklıklar belirlenir.

Bu amaçla (**1,3**) kümesi ile 2, 4 ve 5 numaralı gözlemler arasındaki uzaklığı belirleyelim

Adım 2

Uzaklıklar tablosunda Min*d*(i,j) hücresinin belirlenmesi gerekiyor. Tablo 6.5 incelendiğinde Mind(i,j)=1.41 olduğu görülür. O halde bu değerin ilgili olduğu 1 ve 3 numaralı gözlemeler ele alınır. Bu iki değer birleştirilerek (**1,3**) **kümesi** elde edilir.
Şimdi bu yeni elde edilen kümeye göre uzaklıklar matrisini yeniden gözden geçirmemiz gerekmektedir. Çünkü (**1,3**) kümesi ile diğer gözlemler arasındaki uzaklıkları belirlememiz söz konusudur. Bunun için, söz konusu kümenin elemanları ile diğer gözlemler eşlenerek içlerinden en küçük olanlar, yani birbirine en yakın olan gözlemlerle ilgili olan uzaklıklar belirlenir.
Bu amaçla (**1,3**) kümesi ile 2, 4 ve 5 numaralı gözlemler arasındaki uzaklığı belirleyelim



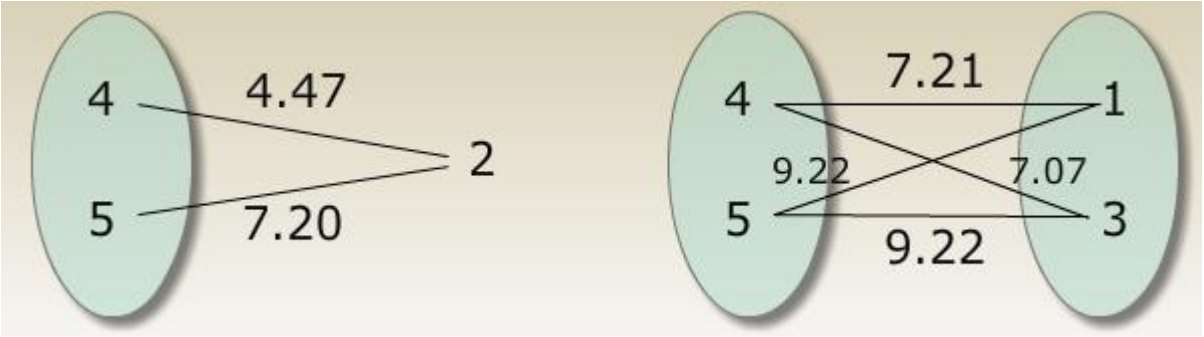
Bu durumda **(1,3) kümesi** ile **2 numaralı** gözlem arasındaki en küçük uzaklık olan **2.83**; **4 numaralı** gözlem ile arasındaki en küçük uzaklık olan **7.07** ve **5 numaralı** gözlem ile arasındaki en küçük uzaklık olan **9.22**değerleri yeni uzaklık değerleri olarak alınır. Bu durumda yeni uzaklıklar tablosu şu şekli alır:

Gözlemler	(1,3)	2	4	5
(1,3)				
2	2.83			
4	7.07	4.47		
5	9.22	7.20	2.24	

Tablo 6.6: Uzaklıklar tablosu

Adım 3

Buradaki uzaklıklar tablosunu göz önüne alalım. Tablo incelendiğinde Mind(i,j)=2.24 olduğu görülür. O halde bu değerin ilgili olduğu **4** ve **5** gözlemleri birleştirilerek bir küme oluşturacaktır.
Elde edilen (**1**,**3**) kümesinin diğer (**1**,**3**) kümesi ve **2** gözlemi ile olan uzaklıklarını belirlemek gerekiyor. Aşağıdaki şekil üzerinde görüldüğü gibi, (**4,5**) kümesi ile **2** numaralı gözlem arasındaki en küçük mesafe **4.47**olduğundan bu mesafe uzaklık tablosunda göz önüne alınır.
Benzer biçimde (**4,5**) kümesi ile (**1,3**) kümesi arasındaki en küçük uzaklık olan **7.07** değeri tabloda yer alır.



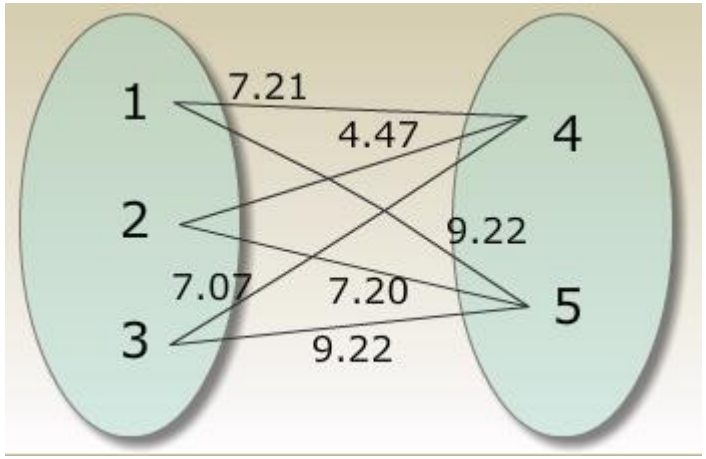
Bu durumda uzaklık tablosu aşağıda belirtilen biçimi alır:

Gözlemler	(1,3)	2	(4,5)
(1,3)			
2	2.83		
(4,5)	7.07	4.47	

Tablo 6.7: Uzaklıklar tablosu

Adım 4

En son uzaklıklar tablosu incelendiğinde Mind(i,j)=2.83 olduğu görülür. O halde bu uzaklık ile ilgili olan **2** gözlemi ile (**1,3**) kümesi birleştirilecektir. Elde edilen (**1,2,3**) kümesi ile (**4,5**) kümesi arasındaki uzaklığı belirlemek için kümeler içindeki her bir değeri eşliyoruz ve aralarında en küçük olanı belirliyoruz.
En küçük uzaklık **4.47** olduğuna göre söz konusu iki küme arasındaki uzaklık olarak bu değer belirlenmiş olur.



Yeni uzaklık değerini de içeren uzaklıklar tablosu şu şekildedir:

Gözlemler	(1,2,3)	(4,5)
(1,2,3)		
(4,5)	4.47	

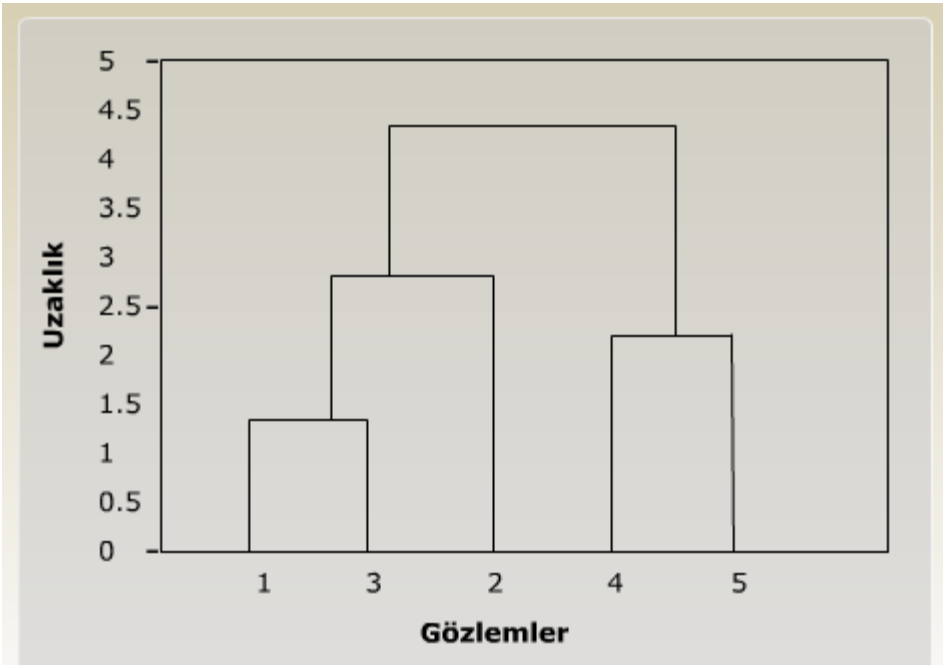
Tablo 6.8: Uzaklıklar tablosu

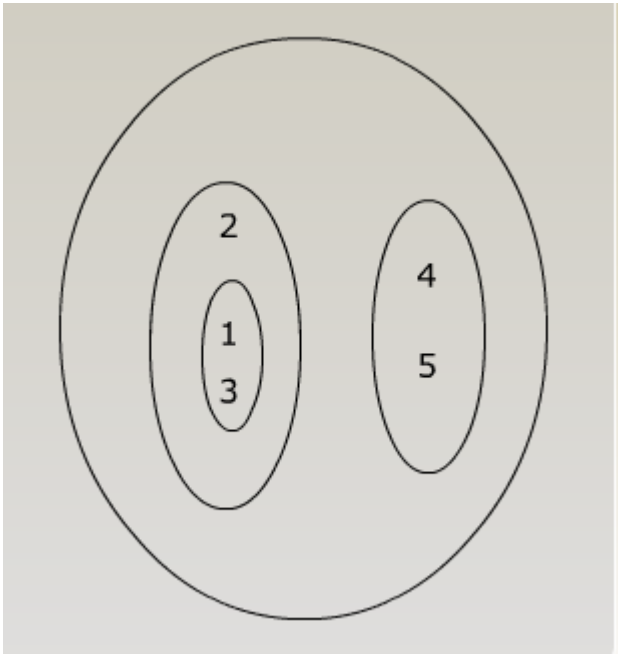
Adım 5

Elde edilen iki küme birleştirilerek sonuç küme elde edilir. Bu küme (1,2,3,4,5) gözlemlerinden oluşan kümedir. Uzaklık düzeyi göz önüne alınarak kümeler şu şekilde belirlenmiştir:

Uzaklık	Kümeler
1.41	(1,3)
2.24	(4,5)
2.83	(1,2,3)
4.47	(1,2,3,4,5)

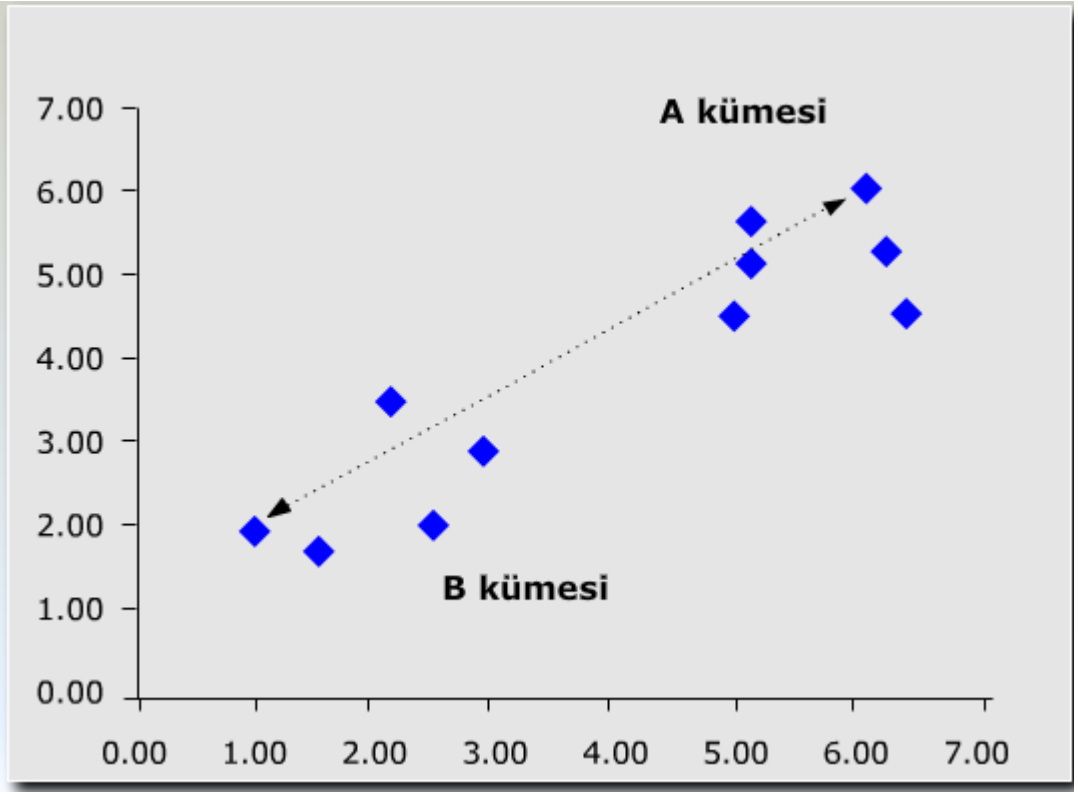
Kümeleme ile ilgili dendogram (solda) ve kümeler (sağda) ise aşağıda belirtildiği biçimdedir.





En Uzak Komşu Algoritması

Bu yönteme "**tam bağlantı kümeleme yöntemi**" adı da verilmektedir. Yöntem en yakın komşu algoritmasına çok benzer. Ancak bu kez kümeler arasındaki uzaklık belirlenirken, iki kümenin birbirine en uzak olan elemanları arasındaki mesafe, iki küme arasındaki uzunluk olarak tayin edilir.



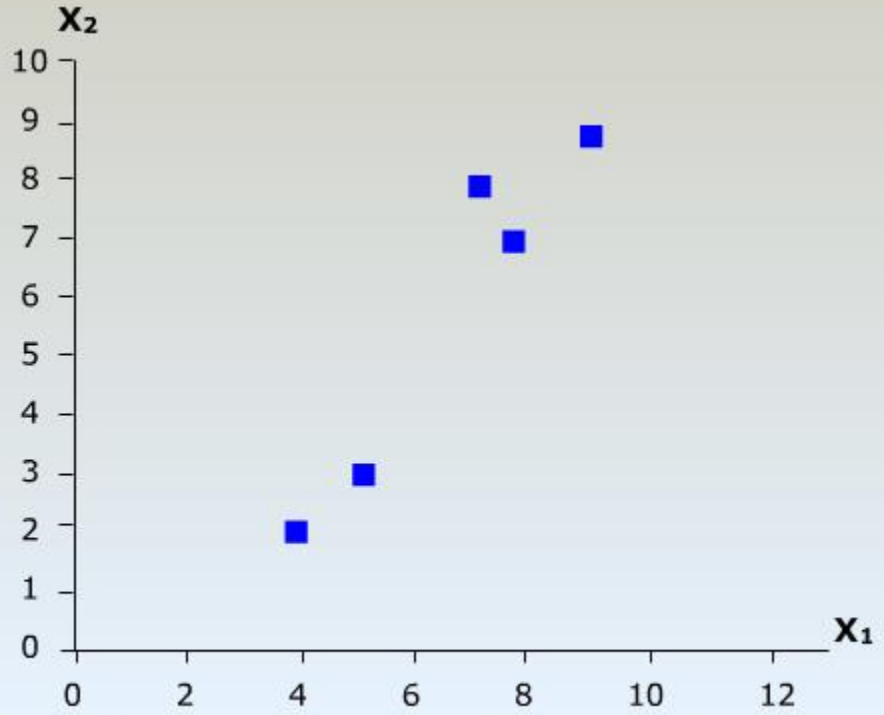
En uzak komşu algoritmasında iki kümenin birbirine en uzak gözlemleri arasındaki uzaklık iki kümenin birbirine olan uzaklığı olarak değerlendirilir.

Uygulama 2

Aşağıdaki gözlem değerlerini göz önüne alalım. Bu kez kümeleme analizini "**en uzak komşu algoritmasına**" göre yapacağız.

Gözlemler	X_1	X_2
1	7	8
2	4	2
3	5	3
4	8	7
5	9	9

Tablo 6.10: Gözlem değerleri.



Şekil 6.11: Gözlem değerlerinin grafik üzerindeki görünümü.

Adım 1

Öncelikle öklid uzaklık bağıntısını kullanarak gözlemler arasındaki uzaklıkları belirliyoruz. Bu amaçla aşağıdaki hesaplamalar yapılır:

$$d(1,2) = \sqrt{(7-4)^2 + (8-2)^2} = 6.71$$
$$d(1,3) = \sqrt{(7-5)^2 + (8-3)^2} = 5.39$$
$$d(1,4) = \sqrt{(7-8)^2 + (8-7)^2} = 1.41$$
$$d(1,5) = \sqrt{(7-9)^2 + (8-9)^2} = 2.24$$
$$d(2,3) = \sqrt{(4-5)^2 + (2-3)^2} = 1.41$$
$$d(2,4) = \sqrt{(4-8)^2 + (2-7)^2} = 6.40$$
$$d(2,5) = \sqrt{(4-9)^2 + (2-9)^2} = 8.60$$
$$d(3,4) = \sqrt{(5-8)^2 + (3-7)^2} = 5.00$$
$$d(3,5) = \sqrt{(5-9)^2 + (3-9)^2} = 7.21$$
$$d(4,5) = \sqrt{(8-9)^2 + (7-9)^2} = 2.24$$

Bu durumda gözlemlere ilişkin uzaklıklar matrisi şu şekilde olacaktır:

Gözlemler	1	2	3	4	5
1					
2	6.71				
3	5.39	1.41			
4	1.41	6.40	5.00		
5	2.24	8.60	7.21	2.24	

Tablo 6.11: Uzaklıklar tablosu

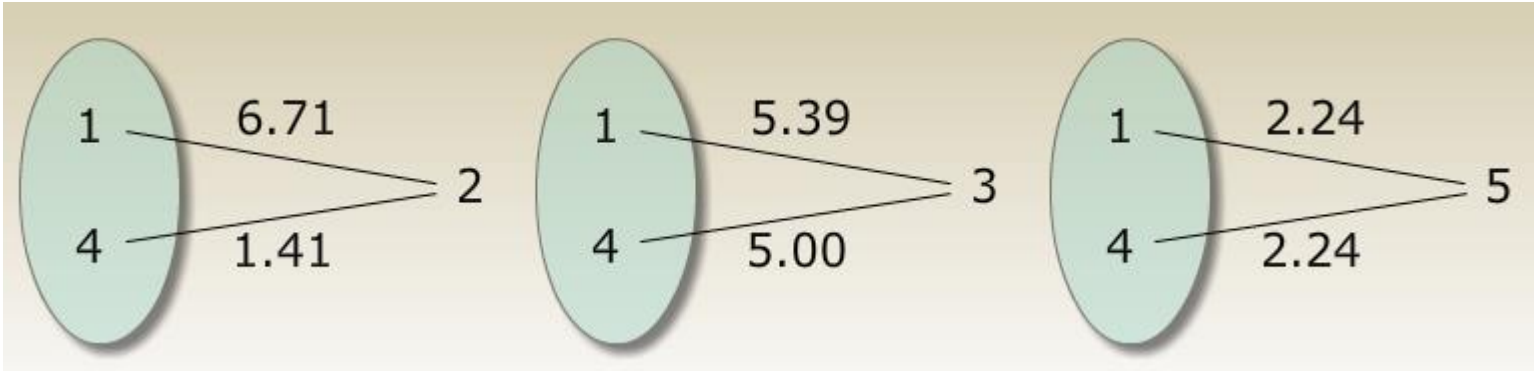
Adım 2

İlk aşamada aynen en yakın komşu algoritmasında olduğu gibi Uzaklıklar tablosunda Min*d*(i,j) hücresinin belirlenmesi gerekiyor. Tablo üzerinde Mind(i,j)=1.41 olduğu görülür. O halde bu değerin ilgili olduğu **1** ve **4**numaralı gözlemler ele alınır. Bu iki değer birleştirilerek (**1,4**) kümesi elde edilir.
Şimdi bu yeni elde edilen kümeye göre uzaklıklar matrisini yeniden gözden geçirmemiz gerekmektedir. Çünkü (1,4) kümesi ile diğer gözlemler arasındaki uzaklıkları belirlememiz söz konusudur. Bunun için, söz konusu kümenin elemanları ile diğer gözlemler eşlenerek içlerinden **birbirine en uzak olan gözlemler** ile olan uzaklıklar belirlenir. Bu amaçla (**1,4**) kümesi ile 2, 3 ve 5 numaralı gözlemler arasındaki uzaklığı belirleyelim.

Adım 2

İlk aşamada aynen en yakın komşu algoritmasında olduğu gibi Uzaklıklar tablosunda Min*d*(i,j) hücresinin belirlenmesi gerekiyor. Tablo üzerinde Mind(i,j)=1.41 olduğu görülür. O halde bu değerin ilgili olduğu **1** ve **4**numaralı gözlemler ele alınır. Bu iki değer birleştirilerek (**1,4**) kümesi elde edilir.

Şimdi bu yeni elde edilen kümeye göre uzaklıklar matrisini yeniden gözden geçirmemiz gerekmektedir. Çünkü (1,4) kümesi ile diğer gözlemler arasındaki uzaklıkları belirlememiz söz konusudur. Bunun için, söz konusu kümenin elemanları ile diğer gözlemler eşlenerek içlerinden **birbirine en uzak olan gözlemler** ile olan uzaklıklar belirlenir. Bu amaçla (**1,4**) kümesi ile 2, 3 ve 5 numaralı gözlemler arasındaki uzaklığı belirleyelim.



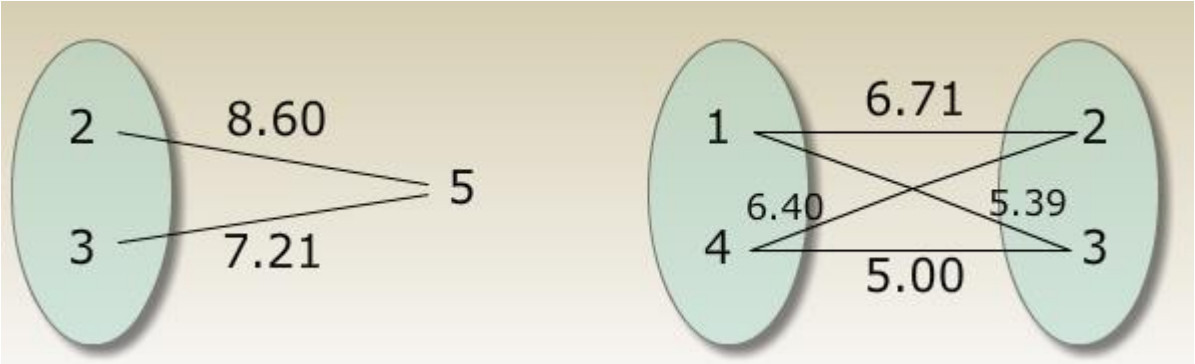
Bu durumda (**1,4**) kümesi ile **2** numaralı gözlem arasındaki en büyük uzaklık olan **6.71**; **3** numaralı gözlem ile arasındaki en büyük uzaklık olan **5.39** ve **5** numaralı gözlem ile arasındaki en büyük uzaklık olan **2.24**değerleri yeni uzaklık değerleri olarak alınır. Bu durumda yeni uzaklıklar t

Gözlemler	(1,4)	2	3	5
(1,4)				
2	6.71			
3	5.39	1.41		
5	2.24	8.60	7.21	

TABLO-6.12: Uzaklıklar tablosu

Adım 3

Tablo 6.12'deki uzaklıklar tablosunu göz önüne alalım. Tablo incelendiğinde Min*d*(i,j)=1.41 olduğu görülür. O halde bu değerin ilgili olduğu **2** ve **3** gözlemleri birleştirilerek bir küme oluşturacaktır.
Elde edilen (**2,3**) kümesinin diğer (**1,4**) kümesi ve **5** gözlemi ile olan uzaklıklarını belirlemek gerekiyor. Aşağıdaki şekil üzerinde görüldüğü gibi, (**2,3**) kümesi ile 5 numaralı gözlem arasındaki en büyük mesafe **8.60**olduğundan bu mesafe uzaklık tablosunda göz önüne alınır. Benzer biçimde (**1,4**) kümesi ile (**2.3**) kümesi arasındaki en büyük uzaklık olan **6.71** değeri tabloda yer alır



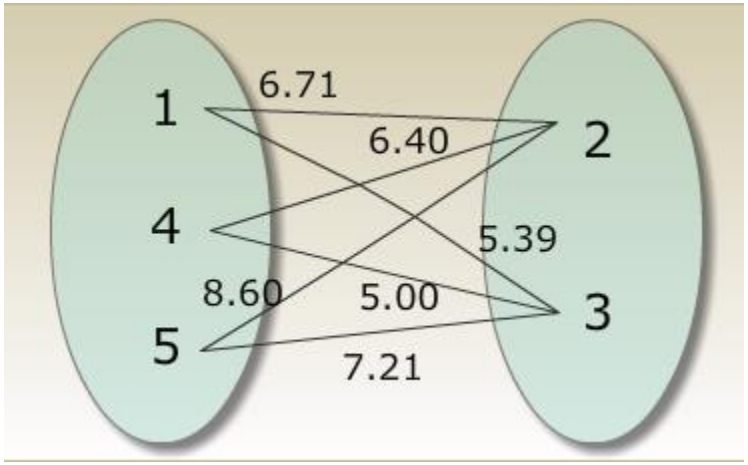
Bu durumda uzaklık tablosu aşağıda belirtilen biçimi alır:

Gözlemler	(1,4)	(2,3)	5
(1,4)			
(2,3)	6.71		
5	2.24	8.60	

Tablo 6.13: Uzaklıklar tablosu

Adım 4

En son uzaklıklar tablosu incelendiğinde Mind(i,j)=2.24 olduğu görülür. O halde bu uzaklık ile ilgili olan **5** gözlemi ile (**1,4**) kümesi birleştirilecektir. Elde edilen (**1,4,5**) kümesi ile (**2,3**) kümesi arasındaki uzaklığı belirlemek için kümeler içindeki her bir değeri eşliyoruz ve aralarında en büyük olanı belirliyoruz.
En büyük uzaklık **8.60** olduğuna göre söz konusu iki küme arasındaki uzaklık olarak bu değer belirlenmiş olur.



Yeni uzaklık değerini de içeren uzaklıklar tablosu şu şekildedir:

Gözlemler	(1,4,5)	(2,3)
(1,4,5)		
(2,3)	8.60	

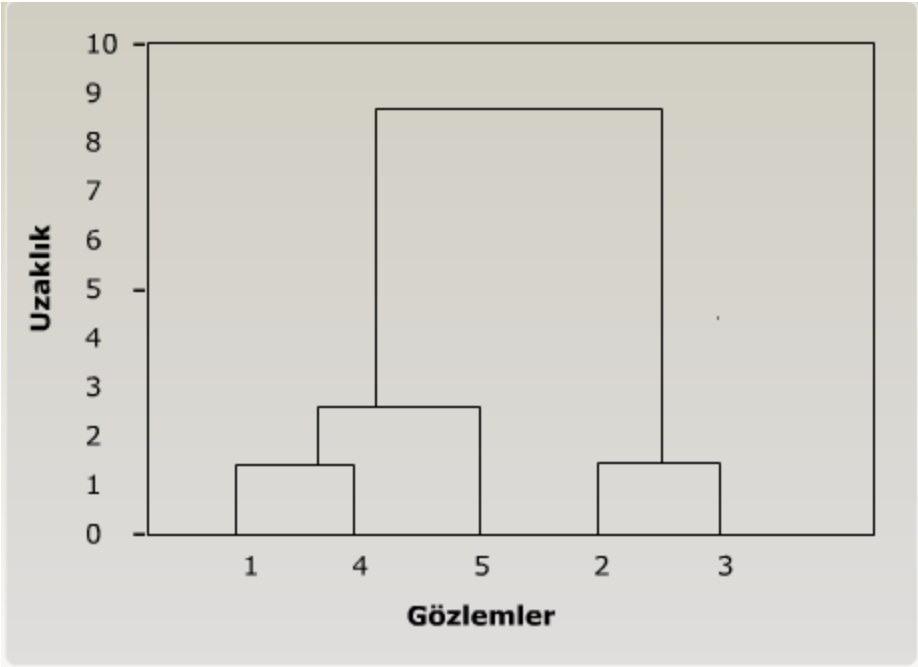
Tablo 6.14: Uzaklıklar tablosu

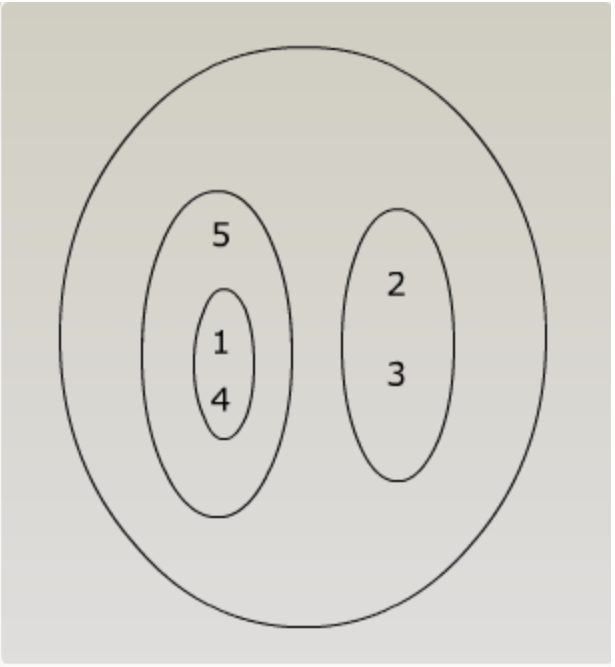
Adım 5

Elde edilen iki küme birleştirilerek sonuç küme elde edilir. Bu küme (**1,2,3,4,5**) gözlemlerinden oluşan kümedir. Uzaklık düzeyi göz önüne alınarak kümeler şu şekilde belirlenmiştir.

Uzaklık	Kümeler
1.41	(1,4)
1.41	(2,3)
2.24	(1,4,5)
8.60	(1,2,3,4,5)

Kümeleme ile ilgili **dendogram** ve **kümeler** ise aşağıda belirtildiği biçimdedir.





Hiyerarşik Olmayan Kümeleme: k-ortalamalar Yöntemi

Hiyerarşik olmayan kümeleme yöntemleri arasında **k-ortalamalar** (*k-mean*) yöntemi önem taşır ve yaygın biçimde kullanılır. Bu yöntemde, daha başlangıçta belli sayıdaki küme içim toplam ortalama hatayı minimize etmek amaçlanır .
N boyutlu uzayda N örnekli kümlerin verildiğini varsayalım. Bu uzay $\{C_1,C_2,...,C_k\}$ biçiminde K kümeye ayrılsın. O zaman $\Sigma n_k=N$ (k=1,2,..,k) olmak üzere C_k kümesinin ortalama vektörü M_k şu şekilde hesaplanır:

$$M_k = \frac{1}{n_k}\sum_{i=1}^{n_k} X_{ik}$$

Burada X_k değeri C_k kümesine ait olan i. örnektir. C_k kümesi için kareli-hata, her bir C_k örneği ile onun **merkezi** (centroid) arasındaki Öklid uzaklıkları toplamıdır. Bu hataya "**küme içi değişme**" adı da verilir. Küme içi değişmeler şu şekilde hesaplanır:

$$e_i^2 = \sum_{i=1}^{n_k}(x_{ik} - M_k)^2$$

K kümesini içeren bütün kümeler uzayı için kare-hata, küme içindeki değişmelerin toplamıdır. O halde söz konusu kare-hata değeri şu şekilde hesaplanır:

$$E_k^2 = \sum_{k=1}^{K} e_k^2$$

Kare-hata kümeleme yönteminin amacı, verilen K değeri için E^2_kdeğerini minimize eden K kümesini içeren bir bölgeyi bulmaktır.

Algoritma

K-ortalama algoritmasına başlamadan önce, **k** küme sayısının belirlenmesi gerekir. Söz konusu **k** değeri belirlendikten sonra her bir kümeye gözlem değerleri atanır ve böylece $C_1,C_2,...,C_k$ kümeleri belirlenmiş olur. Ardından aşağıdaki işlemler gerçekleştirilir:

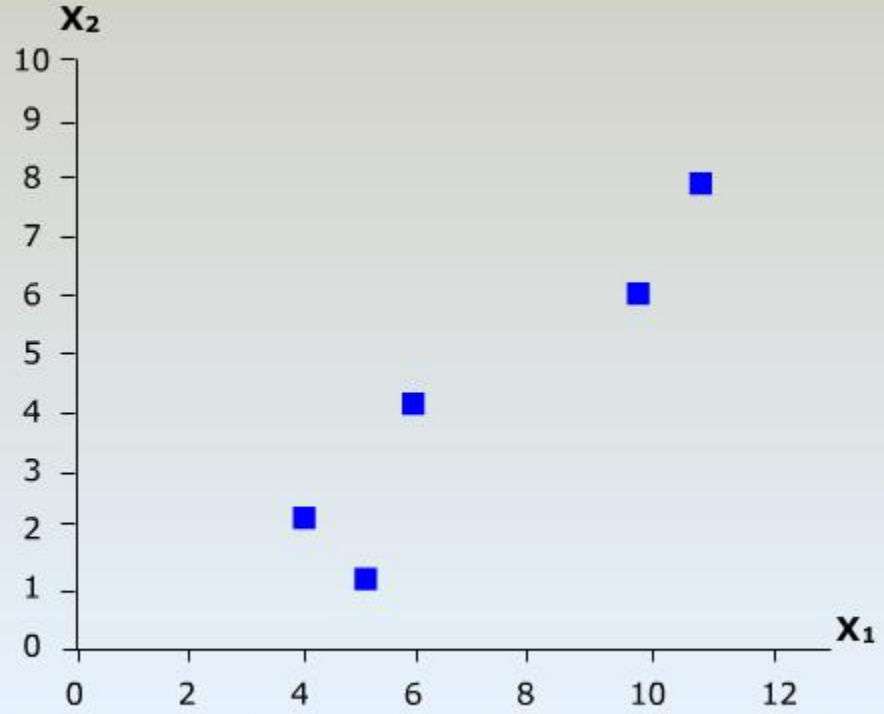
a	Her bir kümenin merkezi belirlenir. Bu merkezler M_1 , M_2 ,..., M_k biçimindedir.
b	e_1 , e_2 , ... , e_k küme içi değişmeler hesaplanır. Bu değişmelerin toplamı olan değeri bulunur.
c	M_k merkez değerleri ile gözlem değerleri arasındaki uzaklıklar hesaplanır. Bir gözlem değeri hangi merkeze yakın ise, o merkez ile ilgili küme içine dahil edilir.
d	Yukarıdaki b ve c adımları, kümelerde herhangi bir değişiklik olmayıncaya dek sürdürülür.

Uygulama 3

Aşağıdaki gözlem değerlerini göz önüne alalım. Bu gözlem değerlerine k-ortalamalar yöntemini uygulayarak kümelemek istiyoruz.

Gözlemler	Değişken1	Değişken2
X_1	4	2
X_2	6	4
X_3	5	1
X_4	10	6
X_5	11	8

Tablo 6.16: Gözlem değerleri.



Şekil 6.17: Gözlem değerlerinin grafik üzerindeki görünümü.

Kümelerin sayısına başlangıçta **k=2** biçiminde karar veriyoruz. Başlangıçta tesadüfi olarak aşağıdaki iki kümeyi belirliyoruz:

$C_1=\{X_1,X_2,X_4\}$

$C_2=\{X_3,X_5\}$

Bu kümeleri de içeren gözlem değerlerini aşağıdaki tablo üzerinde topluca gösteriyoruz.

Gözlemler	Değişken1	Değişken2	Küme üyeliği
X_1	4	2	C_1
X_2	6	4	C_1
X_3	5	1	C_2
X_4	10	6	C_1
X_5	11	8	C_2

TABLO- 6.17: Gözlem değerleri.

Adım 1

a) Bir önceki sayfada belirtilen iki kümenin merkezleri şu şekilde hesaplanır:

$$M_1 = \left\{\frac{4+6+10}{3}, \frac{2+4+6}{3}\right\}$$
$$= \{6.67,\ 4.0\}$$

$$M_2 = \left\{\frac{5+11}{2}, \frac{1+8}{2}\right\}$$
$$= \{8.00,\ 4.50\}$$

Gözlemler	Değişken1	Değişken2	Küme üyeliği
X_1	4	2	C_1
X_2	6	4	C_1
X_3	5	1	C_2
X_4	10	6	C_1
X_5	11	8	C_2

TABLO- 6.17: Gözlem değerleri.

b) Küme içi değişmeler şu şekilde hesaplanır:

$$e_1^2 = \left[(4-6.67)^2 + (2-4.00)^2\right] + \left[(6-6.67)^2 + (4-4.00)^2\right]$$
$$+ \left[(10-6.67)^2 + (6-4.00)^2\right]$$
$$= 26.67$$

$$e_2^2 = \left[(5-8)^2 + (1-4.50)^2\right] + \left[(11-8)^2 + (8-4.50)^2\right]$$
$$= 42.50$$

Bu durumda toplam **kare-hata** şu şekilde hesaplanır:

$E_2=e^2_1+e^2_2 = 26.67+42.50 = 69.17$

c) M_1 ve M_2 merkezlerinden olan uzaklıkların minimum olması istendiğinden aşağıdaki hesaplamalar yapılır. Öklid uzaklık formülü kullanılarak söz konusu mesafeler hesaplanır. Örneğin (M_1,X_1) noktaları arasındaki uzaklık, M_1= {6.67,4.00 } ve X_1={4,2} olduğuna göre şu şekilde hesaplanır:

$$d(M_1, X_1) = \sqrt{(6.67-4)^2 + (4-2)^2}$$
$$= 3.33$$

Bu kez, M_2= {8,4.5 } ve X_1={4,2} olduğuna göre (M_2,X_1) uzaklığı şu şekilde bulunur:

$$d(M_2, X_1) = \sqrt{(8-4)^2 + (4.5-2)^2}$$
$$= 4.72$$

Yukarıdaki işlemlerden şu anlaşılıyor:
X_1 gözlem değerinin M_1 ve M_2 merkezlerine olan uzaklıkları göz önüne alındığında $d(M_1,X_1)<d(M_2,X_1)$ olduğu görülür. Bu durumda M_1 merkezinin X_1 gözlem değerine daha yakın olduğu anlaşılır. O halde X_1C_1olarak kabul edilir. Benzer biçimde yukarıdaki işlemler tüm gözlem değerleri için tekrarlanarak aşağıdaki tablo elde edilir.

Gözlemler	M_1 den uzaklık	M_2 den uzaklık	Küme üyeliği
X_1	$d(M_1,X_1)=3.33$	$d(M_2,X_1)=4.72$	C_1
X_2	$d(M_1,X_2)=0.67$	$d(M_2,X_2)=2.06$	C_1
X_3	$d(M_1,X_3)=3.43$	$d(M_2,X_3)=4.61$	C_1
X_4	$d(M_1,X_4)=3.89$	$d(M_2,X_4)=2.50$	C_2
X_5	$d(M_1,X_5)=5.90$	$d(M_2,X_5)=4.61$	C_2

Tablo 6.18

Bu durumda yeni kümeler şu şekilde olacaktır :

$C_1=\{X_1,X_2,X_3\}$

$C_2=\{X_4,X_5\}$

Adım 2

a) Bir önceki sayfada belirtilen iki kümenin merkezleri şu şekilde hesaplanır:

$$M_1 = \left\{\frac{4+6+5}{3}, \frac{2+4+1}{3}\right\}$$
$$= \{5,\ 2.33\}$$

$$M_2 = \left\{\frac{10+11}{2}, \frac{6+8}{2}\right\}$$
$$= \{10.5,\ 7\}$$

b) Küme içi değişmeler şu şekilde hesaplanır:

$$e_1^2 = \left[(4-5)^2 + (2-2.33)^2\right] + \left[(6-5)^2 + (4-2.33)^2\right]$$
$$+ \left[(5-2.33)^2 + (1-2.33)^2\right]$$
$$= 9.33$$
$$e_2^2 = \left[(10-10.5)^2 + (6-7)^2\right] + \left[(11-10.5)^2 + (8-7)^2\right]$$
$$= 2.50$$

Bu durumda toplam **kare-hata** şu şekilde hesaplanır:
$E_2=e^2_1+e^2_2 = 9.33+2.50 = 11.83$

c) M_1 ve M_2 ve merkezlerinden gözlem değerlerine olan uzaklıklar hesaplanır. X1 gözlem değerinin M1 ve M2 merkezlerine olan uzaklıkları göz önüne alındığında d(M1,X1)<d(M2,X1) olduğu görülür. Bu durumda M1 merkezinin X1 gözlem değerine daha yakın olduğu anlaşılır. O halde X_1C_1 olarak kabul edilir. Benzer biçimde yukarıdaki işlemler tüm gözlem değerleri için tekrarlanarak aşağıdaki tablo elde edilir.

Gözlemler	M_1 den uzaklık	M_2 den uzaklık	Küme üyeliği
X_1	$d(M_1,X_1)=1.05$	$d(M_2,X_1)=8.20$	C_1
X_2	$d(M_1,X_2)=1.94$	$d(M_2,X_2)=5.41$	C_1
X_3	$d(M_1,X_3)=1.33$	$d(M_2,X_3)=8.14$	C_1
X_4	$d(M_1,X_4)=6.20$	$d(M_2,X_4)=1.12$	C_2
X_5	$d(M_1,X_5)=8.25$	$d(M_2,X_5)=1.12$	C_2

Tablo 6.19

Bu durumda yeni kümeler şu şekilde olacaktır:
$C_1=\{X_1,X_2,X_3\}$
$C_2=\{X_4,X_5\}$

Adım 3

Bu durumda yeni kümeler şu şekilde olacaktır:

$C_1=\{X_1,X_2,X_3\}$

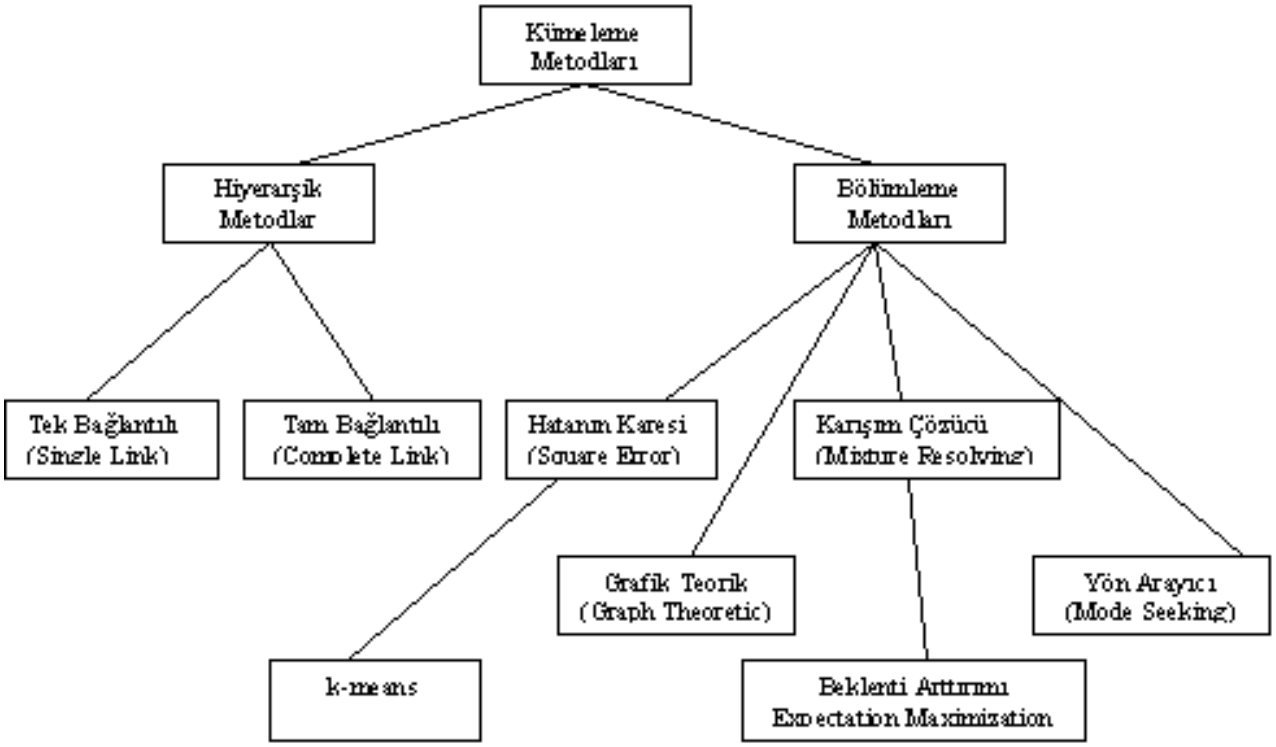
$C_2=\{X_4,X_5\}$

Kümeleme

Verilerin birbirine benzeyen kısımlarının gruplandırılmasına kümeleme adını veriyoruz. Kümeleme çözümlemeleri veri madenciliğinde gemniş bir uygulama alanı bulmuştur. Bu ders kapsamında verilerin birbirine olan uzaklıklarını esas alarak **hiyerarşik** ve **hiyerarşik olmayan** kümeleme biçiminde geliştirilmiş iki algoritma türü anlatılmaktadır.

Hiyerarşik Kümeleme Metotları

En yaygın olarak kullanılan veri madenciliği tekniklerinden biri olan kümeleme analizini gerçekleştirmek için birçok kümeleme metodu geliştirilmiştir. Kümeleme metotları kullandıkları kümeleme yöntemlerine bağlı olarak bir hiyerarşi oluşturmaktadır. Jain ve Dubes tarafından oluşturulan kümeleme metotları hiyerarşisi Şekilde görülmektedir.



Hiyerarşik Kümeleme

- Hiyerarşik küme, bir veri setindeki her bir nesnenin dizideki bir sonraki nesnenin içinde yer aldığı bir nesneler dizisidir.
- Bu dizinin en üst seviyesinde tüm nesneleri içeren tek bir küme ve en alt seviyesinde ise ayrı noktalardan oluşan tekil kümeler yer alır.
- Bu iki seviye arasında kalan her seviyedeki küme,bu küme ve bu kümenin bir alt (veya bir üst) seviyesindeki kümenin birleşimi (veya ayrışımı)dir.
- Hiyerarşik kümeleme metotları, nesnelerin iç içe gruplanma ilişkisini ve gruplanmaların değiştiği benzerlik seviyelerini ağaç yapısı şeklinde gösteren bir dendrogram oluşturma temeline dayanır.

Hiyerarşik Kümeleme

Hiyerarşik kümeleme metotları

i. nesneler arasındaki hiyerarşik ilişkiyi gösteren dendrogram yapısını, nesneleri veya küçük kümeleri birleştirerek yada büyük kümeleri parçalara bölerek oluşturur.

ii. kümelenmiş (agglomerative) veya bölücü (divisive) metotd olarak ikiye ayrılırlar.

iii. Bu durum hiyerarşinin aşağıdan-yukarıya (bottom-up) - birleştirici (merging) - ya da yukarıdan aşağıya (top-down) - bölücü (splitting) - olmasına göre değişir.

iv. Kümelenmiş hiyerarşik kümeleme metodu aşağıdan yukarıya(bottom-up) stratejisini kullanır.

Hiyerarşik Kümeleme

Kümelenmiş hiyerarşik kümeleme metotları

i. aşağıdan yukarıya(bottom-up) stratejisini kullanır.

ii. Bu yöntem de tipik olarak, her nesne başta küme olarak kabul edilerek tekrarlı bir biçimde birleşir ve büyük ve daha da büyük kümeler oluşturur.

iii. En son bütün nesneler sadece bir kümenin içinde olduğunda tekrarlama işlemi durur.

iv. Bu oluşan en büyük ve tek küme hiyerarşinin kökü, en tepesi olur.

v. Birleştirme adımında, birbirine en yakın iki kümeyi bulur (bazı yakınlık ölçülerine göre), ve onları birleştirerek tek bir küme haline getirir.

vi. Bütün kümeler en az bir nesne içerir ve her seferde iki küme birleştirilir. Kümelenmiş metod en fazla *n* tekrar içerebilir.

Hiyerarşik Kümeleme

Bölücü hiyerarşik kümeleme

i. yukarıdan aşağıya(top-down) stratejisini kullanır.

ii. bütün nesneleri bir kümeye yerleştirerek başlar. Bu küme hiyerarşinin kökü olan kümedir.

iii. Daha sonra bu kök küme, daha küçük kümelere bölünür.

iv. Bu işlem özyinelemeli şekilde devam eder.

v. Ne zaman ki kümeler en düşük düzeye gelir yani her kümede sadece bir nesne bulunur ya da birbirine çok benzeyen nesneler içerir o zaman işlem durur.

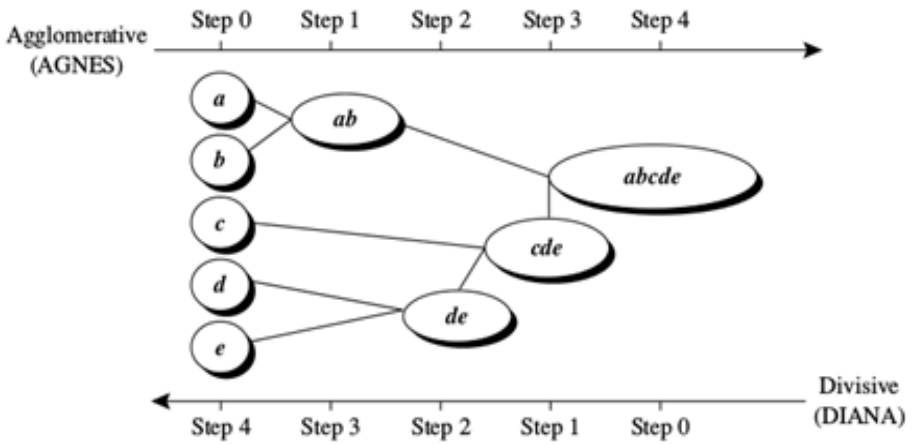
vi. Hem kümelenmiş hem de bölücü hiyerarşik kümeleme metodunda, kullanıcı işlemin bitmesini istediği küme sayısını belirleyebilir.

Birleştirici ve Ayrıştırıcı Kümeleme Algoritmaları

- Birleştirici ve ayrıştırıcı kümeleme algoritmaları en çok kullanılan hiyerarşik kümeleme algoritmalarıdır.
- Bu algoritmalar hiyerarşik kümeleme işlemini en basit tanımıyla gerçekleştirirler.
- Her kümenin dizideki bir sonraki küme içine yerleştirildiği bir küme dizisi oluştururlar. Bu algoritmalar iki grupta incelenir:
  i. Birleştirici Hiyerarşik Kümeleme (AGNES),
  ii. Ayrıştırıcı Hiyerarşik Kümeleme (DIANA).

Birleştirici ve Ayrıştırıcı Kümeleme Algoritmaları

- Aşağıda verilen şekil AGNES(AGlomerative NESting), yani kümelenmiş hiyerarşik metodu, ve DIANA(DIvisive ANAlysis), bölücü kümelenmiş metodu gösteren ve beş tane nesnesi olan bir veri setini gösterir.
- Nesneler {a,b,c,d,e} dir. Başlangıçta, AGNES, kümelenmiş metot, her bir nesneyi bir küme olarak kabul eder ve bazı kriterlere göre adım adım bu kümeler birleşir. Örneğin, C1 ve C2 kümesi, eğer öklit uzaklığına göre birbirine en yakın iki nesneyi bulunduruyorlar ise birleşirler.



Şekil. {a,b,c,d,e} nesneleri ile kümelenmiş ve bölücü hiyerarşik kümeleme

Birleştirici ve Ayrıştırıcı Kümeleme Algoritmaları

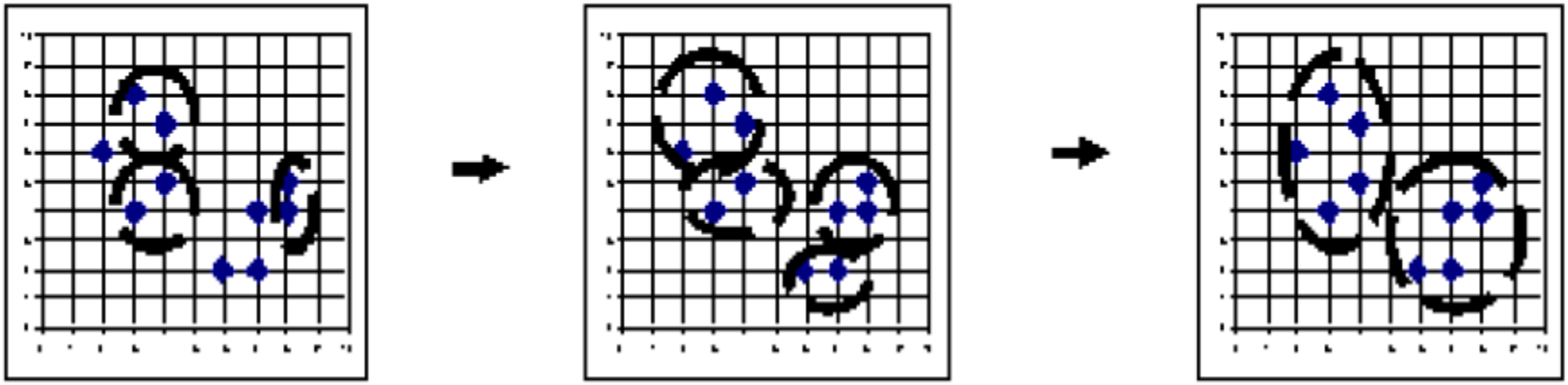
- AGNES (AGglomerative NEsting) algoritması, Kaufman ve Rousseeuw tarafından 1990 yılında sunulmuştur.

- Aşağıdan yukarı doğru çalışan bir inşa yapısı izler. Başlangıçta her nesne ayrı bir küme olarak kabul edilir. Algoritmanın sonraki her adımında bu atomik kümelerden benzer özellik gösterenler birleştirilir.

- Her birleştirme işleminden sonra toplam küme sayısı bir azalır. İstenen sayıda küme elde edildiğinde veya en yakın iki küme arasındaki uzaklık verilen eşik değere ulaştığında birleştirme işlemi sona erer.

Birleştirici ve Ayrıştırıcı Kümeleme Algoritmaları

- Herhangi bir sonlanma koşulu verilmezse kümeleme işlemi tamamlandığında bütün nesneler tek bir kümede toplanır. AGNES algoritmasının çalışma şekli Şekilde görülmektedir.

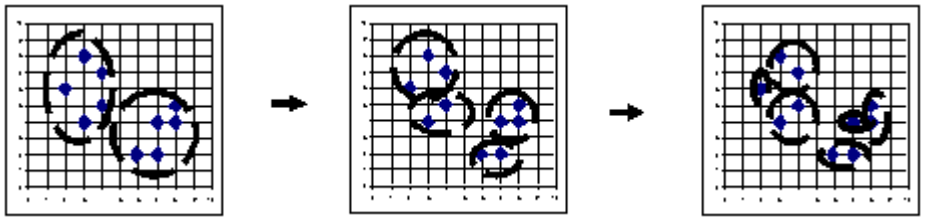


Birleştirici Hiyerarşik Kümeleme Algoritması, AGNES

Birleştirici ve Ayrıştırıcı Kümeleme Algoritmaları

Ayrıştırıcı Hiyerarşik Kümeleme, DIANA

- DIANA (DIvisive ANAlysis) algoritması, Yukarıdan aşağı çalışan bir inşa yapısı izler.
- Başlangıçta veri nesnelerinin tümü tek bir küme olarak kabul edilir.
- Algoritmanın sonraki her adımında kendi aralarında benzerlik oranı en yüksek olan nesneler bir araya getirilip yeni bir küme oluşturularak büyük küme ikiye bölünür.
- Bu işlem her nesne kendi başına bir küme oluşturana kadar veya belli bir sonlandırma koşulu sağlanana kadar devam eder.
- Sonlandırma koşulu istenen sayıda küme elde edilmesi veya en yakın iki küme arasındaki uzaklığın verilen eşik değerin üzerinde olması sağlanır.
- Ayrıştırıcı kümeleme algoritmasının çalışma şekli Şekilde görülmektedir.



Birleştirici ve Ayrıştırıcı Kümeleme Algoritmaları

Dendogram üzerinde gösterim

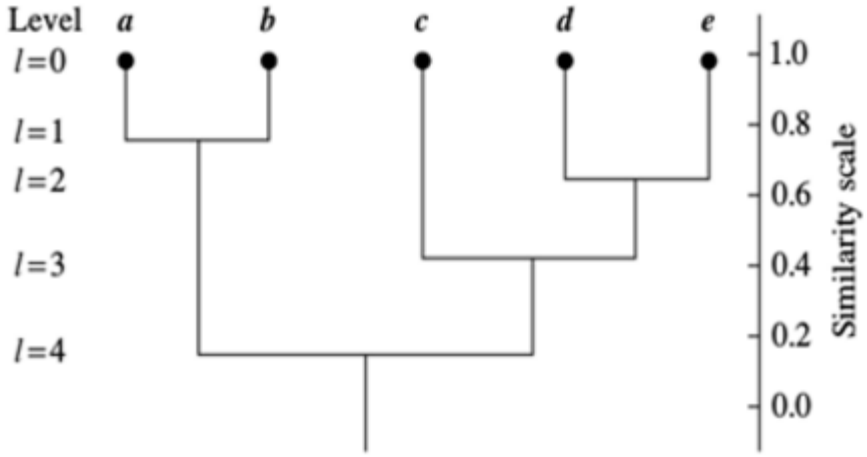
- Bu yöntem tekil-bağlanma (single-linkage) yaklaşımıdır. Bu yaklaşımda, kümeler, her biri ayrı bir küme olarak kabul edilen nesnelerin birleştirilmesiyle oluşturulur. En yakın komşular, yani minimum uzaklığa ya da maksimum benzerliğe sahip olan küme çiftleri bir araya getirilir. Bu işlem bütün nesnelerin tek bir kümeye birleştirilmesine kadar devam eder.

- DIANA, bölücü metod, tersi bir şekilde işlem yapar. Bütün nesneler başlangıçta belirlenen kök kümeyi kullanır. Bu küme, en yakın nesnelerin arasındaki maksimum öklit uzaklığı gibi bazı prensipler dahilinde parçalanır. Bu işlem her küme bir nesne içerene kadar devam eder.

Birleştirici ve Ayrıştırıcı Kümeleme Algoritmaları

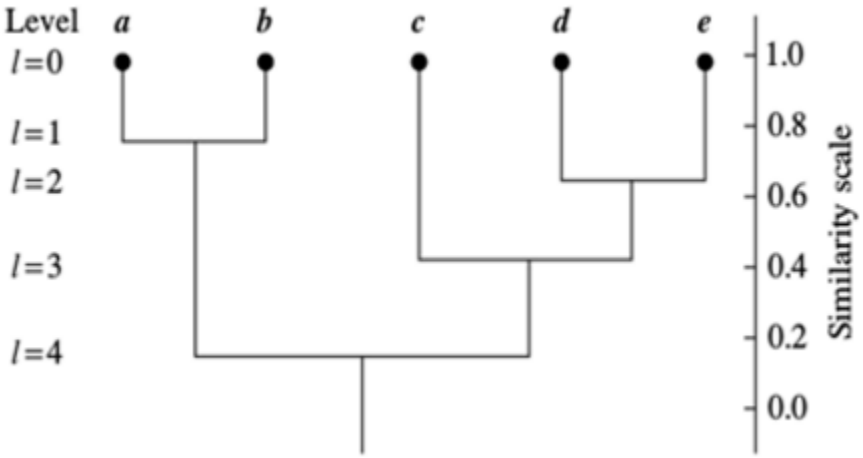
Dendogram üzerinde gösterim

- Hiyerarşik kümeleme işlemini göstermek amacıyla genellikle dendrogram adı verilen ağaç türü kullanılır. Bu ağaç hiyerarşik kümeleme tekniğindeki sonuçları adım adım görselleştirmeye yarar.
- Aşağıdaki şekil bulunan beş tane nesnenin dendrogramıdır. l = 0, beş nesnenin de tekli küme olduğu seviye 0 dır. l = 1 olduğu durumda a ve b birleşmiş ilk kümeyi oluşturmuşlardır. Kümeler arasındaki benzerliği göstermek adına dikey aksı kullanabiliriz. Örneğin, {a,b} ve {c,d,e} gruplarının benzerlikler yaklaşık olarak 0.16 olduğunda iki küme birleşerek tek bir küme olmuştur.



Birleştirici ve Ayrıştırıcı Kümeleme Algoritmaları

- Bölücü metotdaki zor olan kısım, büyük bir kümeyi nasıl bir çok kümeye böleceğimizdir.
- Örneğin, n tane nesne içeren bir kümeyi 2^{n-1}-1 olasılıklı şekilde bölebiliriz.
- n sayısı arttığı taktirde hesaplama süresine üssel olarak artacaktır.
- Sonuç olarak, bölücü metod bölümlerken sezgi(heuristic) kullanır ki, bu durum doğru olmayan sonuçlar ortaya koyabilir.
- Performans artışı için, bölücü metot verdiği bölümleme kararları hakkında tekrar bir hesaplama yapmaz.
- Yani bir kere bölümlenen bir küme tekrar düşünülmez. Bölücü metottaki zorluklar nedeni ile, kümeleme metotları bölücü metotlardan çok daha fazladır.



Birleştirici ve Ayrıştırıcı Kümeleme Algoritmaları

- Literatürde en çok kullanılan hiyerarşik kümeleme algoritması, birleştirici hiyerarşik kümeleme algoritmasıdır.
- Literatürde hiyerarşik algoritmaların farklı örnekleri mevcuttur. Bu algoritmalar kullandıkları benzerlik formülleri ve mevcut kümeler ile oluşturulan yeni kümeler arasındaki benzerlik değerini güncelleme yöntemlerinde farklılık göstermektedir.
- Kullanılan yöntemler genel olarak verilen üç ana metodun benzer varyasyonlarıdır: merkez nokta (centroid veya medoid) tabanlı metotlar, bağlantı tabanlı (linkage-based) metotlar, varyans veya hatanın kareleri toplamı.
- Bu yöntemler, kümeleme işleminin her adımında hangi kümelerin birleşeceğini veya hangi kümenin bölünerek yeni kümeler oluşturacağını belirlemektedir. Merkez tabanlı yaklaşımlarda kümeler kümenin merkezinde bir nokta olan “merkez nokta” ile gösterilir.

Birleştirici ve Ayrıştırıcı Kümeleme Algoritmaları

- İki küme arasındaki uzaklık bu kümelerin merkez noktaları arasındaki uzaklığa eşittir. Merkez tabanlı kümeleme yaklaşımları şekilsiz veya değişik boyutlardaki kümelerin bulunmasında k-means ve k-medoids gibi hiyerarşik olmayan metotlar gibi başarısız olmaktadır. Merkez tabanlı kümeleme metodunun uzaklık formülü aşağıda görülmektedir.

$$d_{mean}(C_i, C_j) = |m_i - m_j|$$

- Ci, Cj : kümeler,
- mi, mj : kümelerin ortalama değerleri.
- Bağlantı tabanlı yaklaşımlar temel olarak üç bölümde incelenir

Birleştirici ve Ayrıştırıcı Kümeleme Algoritmaları

- Bağlantı tabanlı yaklaşımlar temel olarak üç bölümde incelenir

1) Tek bağlantılı kümeleme metodu (Single Linkage)

- Örneklem nokta yoktur,
- Küme, içindeki tüm veri nesneleri ile ifade edilir,
- İki küme arasındaki benzerlik iki kümedeki tüm örüntü çiftleri arasındaki minimum uzaklık değeri ile ölçülür,
- En yakın komşu metodu olarak da adlandırılır,

Birleştirici ve Ayrıştırıcı Kümeleme Algoritmaları

- Tek bağlantılı kümeleme metodunun uzaklık formülü aşağıda görülmektedir

$$d_{min}(C_i, C_j) = \min_{p \in Ci, p' \in Cj} |p - p'|$$

- Ci, Cj : kümeler,
- |p - p'| : p ve p' noktaları arasındaki uzaklık.
- Tek bağlantılı kümeleme metodu şekilsiz ve değişik boyutlardaki kümeleri bulabilir,
- Ayrık olmayan kümelerin bulunmasında başarısız bir metottur ve gürültülü ve istisna verilere karşı hassastır. Bu metodun dağınık ve uzamış kümeler oluşturma eğilimi vardır.

Birleştirici ve Ayrıştırıcı Kümeleme Algoritmaları

2) Tam bağlantılı kümeleme metodu (Complete Linkage)

- İki küme arasındaki benzerlik iki kümedeki tüm örüntü çiftleri arasındaki maksimum uzaklık değeri ile ölçülür,
- En uzak komşu metodu olarak da adlandırılır,
- Tam bağlantılı kümeleme metodunun uzaklık formülü aşağıda görülmektedir.

$$d\max(C_i, C_j) = \max_{p \in Ci, p' \in Cj} |p - p'|$$

- Ci, Cj : kümeler,
- |p - p'| : p ve p' noktaları arasındaki uzaklık.
- Tam bağlantılı kümeleme metodu yoğun ve sıkıca bağlı kümeleri bulabilir,
- Tam bağlantılı kümeleme metodu gürültülü ve istisna verilere karşı daha az hassastır,
- Büyük kümeleri bölebilir,
- Konveks şekilli kümeleri bulamayabilir

Birleştirici ve Ayrıştırıcı Kümeleme Algoritmaları

3) Ortalama bağlantılı kümeleme metodu

- İki küme arasındaki benzerlik iki kümedeki tüm örüntü çiftleri arasındaki uzaklıkların ortalama değerine eşittir,
- -Ortalama bağlantılı kümeleme metodunun uzaklık formülü aşağıda görülmektedir.

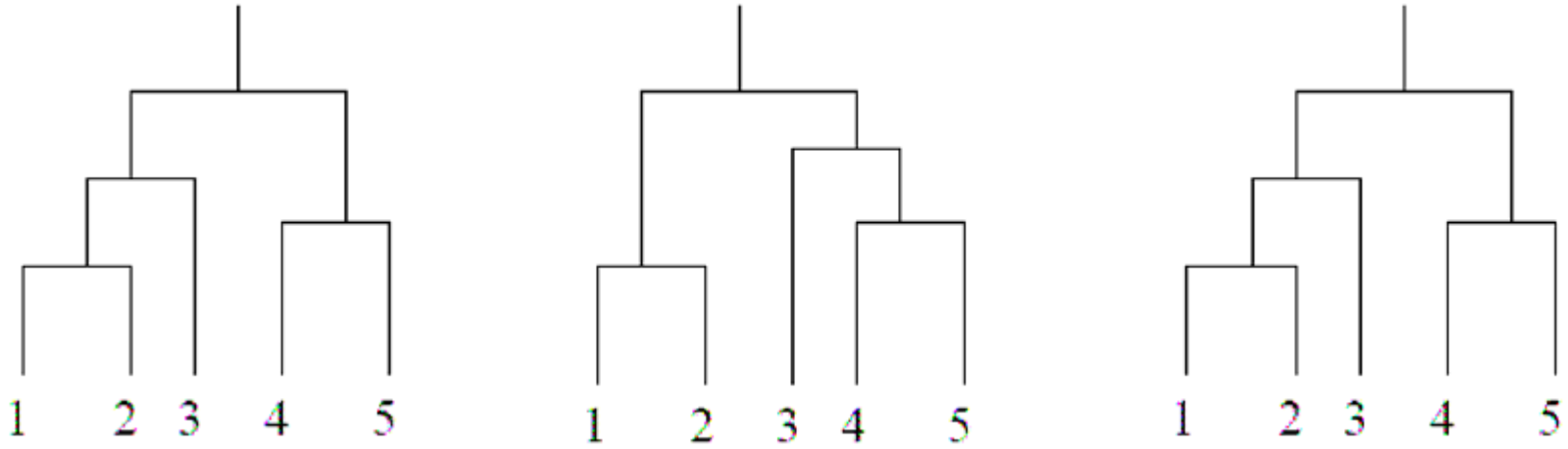
$$davg(C_i, C_j) = 1/(n_i, n_j) \sum_{p \in Ci} \sum_{p' \in Cj} |p - p'|$$

- Ci, Cj : kümeler,
- |p - p'| : p ve p' noktaları arasındaki uzaklık,
- ni, nj : kümelerdeki eleman sayısı.
- Farklı bağlantılı kümeleme algoritmaları ile elde edilen dendrogram yapıları Şekilde görülmektedir.

Birleştirici ve Ayrıştırıcı Kümeleme Algoritmaları

- Farklı bağlantılı kümeleme algoritmaları ile elde edilen dendrogram yapıları Şekilde görülmektedir.

	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Birleştirici ve Ayrıştırıcı Kümeleme Algoritmaları

- Hiyerarşik metotlar kümeler arasındaki benzerliğin ölçülmesi için kullanılan uzaklık formüllerine bağlıdırlar. Bu nedenle veri nesnelerinin bir kısmında doğru ölçekleme yapamazlar.
- Hiyerarşik kümeleme metotları biyolojik bilimlerde sıkça kullanılan alem, filum, cins, tür gibi taksonomilere benzerliği nedeniyle bu alanda etkili bir biçimde kullanılabilmektedir. Hiyerarşik metotlar her tür veri tipine uygulanabilmektedir,
- Hiyerarşik metotların diğer bir çekici özelliği bu metotların belirli bir küme sayısına bağlı olmamasıdır. Giriş parametresi olarak istenen küme sayısını belirten ***k*** değerinin verilmesine gerek yoktur. Tersine dendrogramın uygun bir seviyede kesilmesiyle istenen sayıda küme elde edilebilir,
- Hızlıdır
- Çekirdek noktaların seçilmesine gerek yoktur,
- Benzerlik ve uzaklık seçimi ve idare edilmesi kolaydır,
- Sunumu kolaydır,
- Hiyerarşik metotlar, bölümleyici metotlara oranla daha gerçek sonuçlar vermektedir.

Birleştirici ve Ayrıştırıcı Kümeleme Algoritmaları

AGNES ve DIANA Algoritmalarının Dezavantajları:

- Yanıltıcı olabilir,
- İstisna noktaların etkisi çok fazla olabilir,
- Çok büyük veri setlerinde başarılı sonuçlar elde edilmez,
- Birleştirme ve ayrıştırma işlemleri gerçekleştikten sonra değiştirilip geri alınamamaktadır.

DBSCAN: Yüksek Yoğunluklu Birbirine Bağlı Bölgeler Esaslı Kümeleme

- “Yoğunluk esaslı kümeleme de yoğun bölgeleri nasıl buluruz ? “ , Bir nesnenin - o - yoğunluğu (density) , o nesnesine yakın olan diğer nesneler ile ölçülür.

- DBSCAN (Density-Based Spatial Clustering of Applications with Noise) yoğun komşulukları olan kümeleri bulurki bunlara çekirdek nesneleri(core objects)denir.

- “DBSCAN algoritması bir nesnenin komşuluğunu nasıl ölçümler ?” Kullanıcı tarafından bir nesnenin komşuluk yarıçapı belirlenir $\in>0$. Bu bütün nesneler içinde geçerlidir. Yani $\in$ değeri en büyük komşuluk yarıçapıdır.

DBSCAN: Yüksek Yoğunluklu Birbirine Bağlı Bölgeler Esaslı Kümeleme

- Komşuluk yoğunluğu, komşuluk alanındaki nesnelerin sayısı ile ölçülebilir.
- Bir komşuluğun yoğun olup olmadığına karar vermek için, DBSCAN bir başka kullanıcı tarafından girilen parametre kullanır, ***MinPts***, yoğun alanlardaki minimum yoğunluğu ifade eder.
- Bir veri noktasının çekirdek nokta olması için komşuluk bölgesinde bulunması gereken en az veri noktası sayıdır.
- DBSCAN algoritması, veri noktalarını uzayda oluşturdukları çeşitli yoğunluklardaki bölgelere göre kümelere ayırmaktadır.
- Yoğun bölgeler kümeleri oluştururken, sıradışı ve gürültülü verilerin oluşturduğu seyrek bölgeler tespit edilerek kümelere alınmaz.
- Şekilsiz yada farklı şekillerdeki kümelerin bulunmasında etkili bir algoritmadır.
- En büyük dezavantajı kümelerin yoğunluğunun tanımlanmasında kullanılan giriş parametrelerine karşı duyarlı olmasıdır.

DBSCAN: Yüksek Yoğunluklu Birbirine Bağlı Bölgeler Esaslı Kümeleme

- Bu algoritma, nesnelerin komşuları ile olan mesafelerini hesaplayarak belirli bir bölgede önceden belirlenmiş eşik değerden daha fazla nesne bulunan alanları gruplandırarak kümeleme işlemini gerçekleştirmektedir.
- *public DBSCANalgoritması(DataPoint[] points, Distance distance, double epsilon, int minPoints);*

DBSCAN algoritması parametreleri:

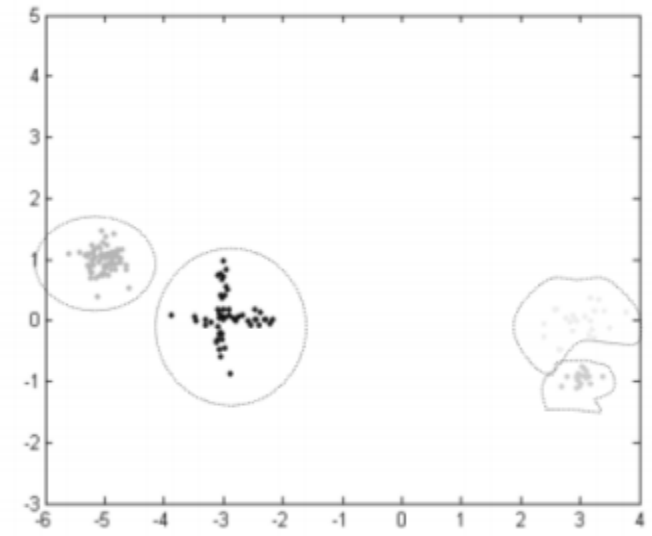
i. Points: Kümeleme yapmak istediğimiz veri setini ifade eder.

ii. Uzaklık ölçümleri: Ne kadar uzaklık için kümeleme yapılacağı bu parametre ile belirlenir

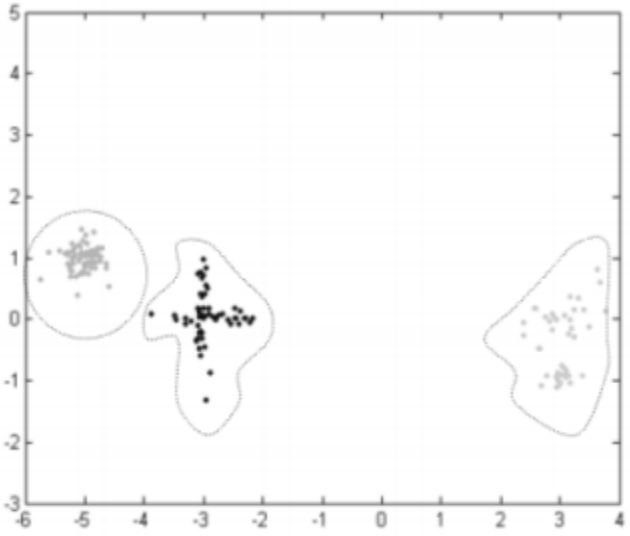
iii. Epsilon: Genellikle çok küçük pozitif bir sayıyı ifade eden parametredir. Bu değer veri noktası p'nin epsilon komşuluk değerinin belirlenmesi için kullanılır. P noktası epsilon değerinden küçük ya da değerine eşit uzaklıktaki noktalar ile komşudur. Bu yüzden "epsilon komşuluğu" şeklinde ifade edilebilir.

iv. ***Minpoints***: Eğer bir nokta bir kümeye ait fakat çekirdek noktalardan biri değilse bu durumda sınır(border) noktalardan biri olmaktadır.

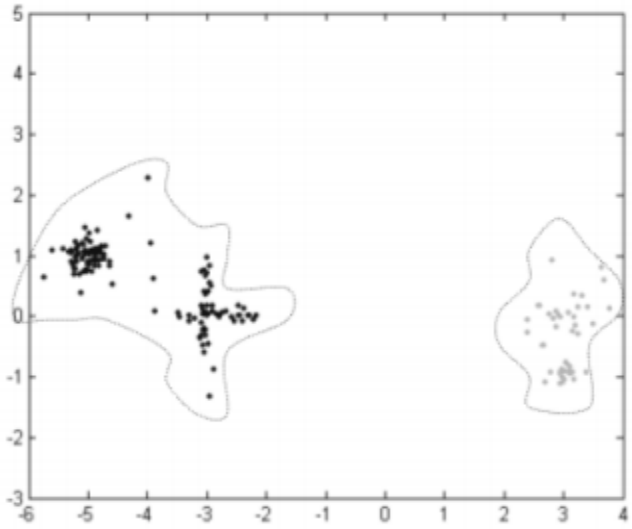
DBSCAN: Yüksek Yoğunluklu Birbirine Bağlı Bölgeler Esaslı Kümeleme



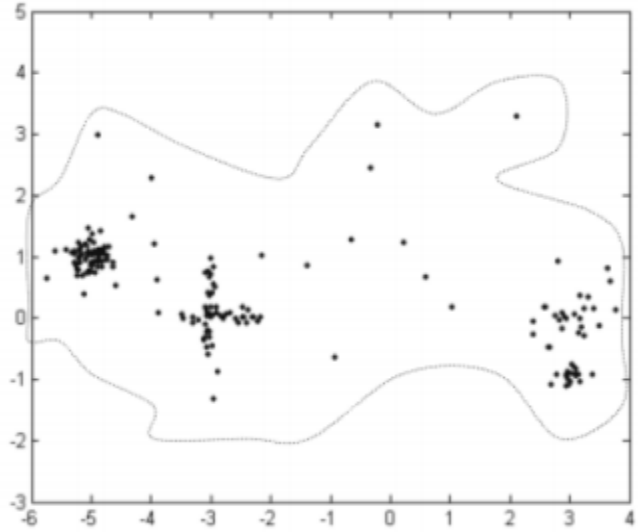
(c) Eps=0.4 ve MinPts=4



(d) Eps=0.6 ve MinPts=6



(e) Eps=0.8 ve MinPts=3



(f) Eps=3 ve MinPts=2

DBSCAN algoritmasında farklı ***k*** değerleri için elde edilen kümeler

DBSCAN: Yüksek Yoğunluklu Birbirine Bağlı Bölgeler Esaslı Kümeleme

Algoritma çalışma mantığını adım adım incelemek gerekirse;

1) Rastgele daha önce işlem yapılmamış bir nokta seçilir.
2) Seçilen bu rastgele noktanın epsilon uzaklığı içerisindeki komşuları bulunur.
3) Eğer komşu sayısı minpoints sayısına eşit ya da büyük ise bir küme oluşturulur; minpoint sayısından küçük ise bu nokta gürültü olarak işaretlenir. Fakat ileriki safhalarda gürültü olarak işaretlenen bu nokta başka bir noktanın oluşturabileceği kümeye dahil edilebilir.
4) Bir nokta bir kümeye dahil edildi ise bu durumda tüm epsilon komşuları da bu kümeye dahil edilir. Tabii ki bu eklenenler için de geçerlidir. Bu işlem eklenecek nokta kalmayana kadar devam ettirilir ve böylece kümenin tamamı oluşturulmuş olur.
5) Yeni üzerinden geçilmemiş rastgele bir nokta seçilir ve döngü tekrarlanır.

DBSCAN: Yüksek Yoğunluklu Birbirine Bağlı Bölgeler Esaslı Kümeleme

DBSCAN kümeleri nasıl buluyor ? ”

1) DBSCAN algoritmasında önce bütün noktalar gezilmedi (unvisited) olarak işaretlenir,
2) ardından rastgele bir noktadan başlanarak tüm noktalar kontrol edilir.
3) Eğer nokta, önceden bir demete eklendiyse işlem yapılmadan sonraki noktaya geçilir. Aksi takdirde, noktanın komşuluğundaki noktalar bulunur.
4) Komşu sayısı MinPts’den küçükse **gürültü** olarak işaretlenir ve sonraki noktaya geçilir.
5) Komşu sayısı MinPts’den büyük veya MinPts’ye eşitse bir demet oluşturulur ve demete bu nokta ve komşuları eklenir.
6) Sonra önceden bir demete eklenmemiş her bir komşu için komşuluğu araştırılarak onun komşuları bulunur.
7) Komşuluğu araştırılan noktaların komşu sayıları MinPts’den büyük veya MinPts’ye eşitse demete eklenir. Bu işlemler eklenecek nokta kalmayana dek devam eder. Sonra bir diğer kümeyi bulmak için veri kümesinden gezilmemiş başka bir nokta seçilerek döngü tekrarlanır. Bütün nesneler gezilene kadar işlem devam eder.

Küme Sayısına Karar Verme

1) Veri kümelerinde “doğru” sayıda kümeye karar vermek önemlidir, sadece bazı kümeleme algoritmalarının örneğin k-ortalama gibi parametre beklediği için değil ayrıca uygun sayıdaki küme sayısı daha uygun bir kümeleme analizi sağlar.

2) Sıkıştırılabilirlik ve kesinlik arasında iyi bir denge yakalamayı sağlar. İki tane uç durum düşünelim.

3) Eğer bütün veri setini küme olarak atamak istersek ne olur ? Bu durum veri özetlemeyi maksimum hale getirirken, kümeleme analizinden hiçbir sonuç alınamayacaktır. Diğer taraftan, veri setindeki her nesneyi bir küme olarak aldığımızda en iyi kümeleme durumunu yakalarız.

4) K-ortalama gibi bazı metodlarda bu durum en iyi durumdur. Fakat, her nesnenin bir küme olduğu verilerin özetlemesi mümkün değildir.

5) Küme sayısına karar vermek bir bakıma kolaydır, çünkü “doğru” rakam belirsizdir.

6) Doğru rakam, genellikle veri kümesinin dağılımına, şekline ve kullanıcının ne çözünürlükte bir cevap beklediğine bağlıdır.

7) Doğru küme sayısını bulabilmek için bir sürü yol vardır. Bazı popüler ve efektif metodları inceleyelim.

Küme Sayısına Karar Verme

1) Basit bir metod, küme sayısını $\sqrt{\frac{n}{2}}$ olarak yapmak. N burada nesne/nokta sayısıdır. Yani, her kümenin $\sqrt{2n}$ noktası olur.

2) Dirsek metodu (elbow method) küme sayısının arttırılması gözlemine dayanır ve her kümenin küme içi toplamını azaltmaya yardımcı olur. Çünkü daha çok küme birbirine daha çok benzeyen nesneler içeren grupları yakalar.

3) Fakat, eğer çok fazla küme oluşmuş ise, küme dahilindeki bu toplam azalma etkisini yitirebilir. Bunun nedeni birleşik olan kümeyi ayırmadan dolayıdır. Yani sonuç olarak, sezgisel olarak küme sayısı için doğru numarayı seçebilmek, küme içi toplam varyansını gösteren eğimde değişim gösteren nokta bizim için önemlidir.

4) Teknik olarak, verilen sayı, k > 0, k kadar verilen kümeyi k-ortalama veya benzeri bir algoritma kullanarak ve küme içi toplamı hesaplayıp, var(k) biçimlendirebiliriz. k ya bağlı olan bu var değişkeninin eğimini görselleştirebiliriz. Eğimdeki ilk veya en önemli değişim noktası “doğru” noktadır olarak adlandırılabilir.

Küme Sayısına Karar Verme

- Daha gelişmiş metodlarda küme sayısına karar verilirken kritik bilgilerden veya teorik yaklaşımlardan yararlanılır.

- “Doğru” küme sayısına karar verebilmek için bir sınıflandırma tekniği olan çapraz doğrulama (cross validation) kullanılabilir. İlk olarak, verilen veri kümesi D’yi m parçaya böleriz. Daha sonra, kümeleme modeli yaratmak için m -1 parça kullanılır ve kalan kısımlar kümelemenin kalitesini test etmek için kullanılır. Örneğin, test kümesindeki her nokta için, en yakın merkezi bulabiliriz. Dolayısıyla, test kümesindeki her noktanın uzaklığının karesinin toplamını ve en yakın merkezleri ölçerek kümeleme modelinin test kümesine uyup uymadığı için kullanabiliriz. $k > 0$ olacak şekilde herhangi bir sayı için, bu işlemi k kümeye ulaşmak için **m** kez tekrar ederiz. Ortaya çıkan ortalama kalite ölçüsü, genel kalite ölçüsü olur ve bu işlemi değişik k değerleri içinde yapıp çıkan sonuçları birbiriyle kıyaslarız. Sonuçta, verimiz için en uygun küme sayısını buluruz.

Kümelemenin Kalitesini Ölçmek (Measuring Clustering Quality)

- Farzedelim ki, bir veri kümesine kümeleme yöntemi uyguladınız. Büyük ihtimalle işlem öncesinde küme sayısını da belirlemeye çalıştınız. Oluşmasını istediğiniz küme sayısından sonra bir ya da birden çok kümeleme metodunu veri kümenize uyguladınız. “Uyguladığınız bu kümeleme metodu ne kadar başarılı oldu ve uyguladığımız değişik kümeleme metodlarının başarılarını nasıl kıyaslarız ? ”

- Kümeleme işleminin kalitesini ölçmek için bazı metodlar mevcuttur. Bu metodlar genellikle bölge araştırması bağlı olarak iki gruba ayrılırlar. Bölge araştırması olarak adlandırdığımız parametre uzman kişiler tarafından yapılan ideal kümeleme işlemidir.

- Eğer bölge araştırması yapılabilir ise, yapı dışı metodlar (Extrinsic methods) tarafından kullanılabilir. Eğer bölge araştırması yapılamaz ise, yapı içi metodlar (instrinsic methods) kullanılır. Yapı içi metodları kümelerin ne kadar iyi ayrıldığına göre kümeleri değerlendirir. Bölge araştırması, “kümelerin etiketlenmesini” gözetleme olarak düşünülebilir. Bundan dolayı, yapı dışı metodlar ayrıca gözetimli metodlar (supervised), yapı içi metodlar ise gözetimsiz metodlar(unsupervised) metodlar olarak bilinir.

Kümelemenin Kalitesini Ölçmek (Measuring Clustering Quality)

- **Yapı Dışı metodlar (Extrinsic Methods)**
- Bölge araştırması mümkün olduğu durumda, kümelere değer vererek karşılaştırma yapabiliriz. Burada asıl görev yapı dışı metodun bir skor belirlemesidir, Q(C,Cg), burada C kümelemeyi, Cg ise bölge araştırmasını ifade ediyor.

- Genel olarak, Q değeri kümeleme işleminde kalite için, eğer aşağıdaki dört durum sağlanırsa, efektif olur. Bahsedilen dört kriter :

(i) Küme homojenliği: Kümeler ne kadar saf olursa, kümeleme işlemi o kadar başarılı olur. Varsayalım ki, bölge araştırması D adında bir veri kümemiz olduğunu, ve L1 , . . . , Ln kategorilerimiz olduğunu söylesin. C1 kümeleme analizini düşünelim ve iki tür nesne türü içersin. Bir de C2 kümelemesini düşünelim, C1 ile tıpatıp aynı sadece kümeleri nesne türlerine göre ayrılmış şekilde olsun. Bu durumda C2'nin Q değeri C1 'den büyük çıkar yani C2 daha iyi kümeleme işlemidir denebilir. Q(C2, Cg) > Q(C1, Cg).

Kümelemenin Kalitesini Ölçmek (Measuring Clustering Quality)

Yapı Dışı metodlar (Extrinsic Methods)

(ii) Kümenin bütünlüğü: Bu kısım küme homojenliği bölümünün benzeri olan bir kısım. Küme bütünlüğü aslında şu demektir. Eğer aynı kategoride olan iki nesne var ise, o zaman onlar aynı küme içinde olmalıdırlar. Örneğin birC1 kümelemesi düşünelim, aynı kategoriye ait nesneleri iki kümede toplasın. Ayrıca bir tane de C2 kümelemesi düşünelim, C1 ile aynı olan fakat sadece kümeleme işleminde iki yerine tek küme kullansın. Bu durumda kümeleme işleminin ölçüsünü sembolize eden Q değeri, kümenin bütünlüğü ilkesine göre C2 kümelemesinde daha yüksek değer alır. Q(C2, Cg) > Q(C1, Cg).

(iii) **Yamalı çanta:** Bir çok senaryo da, çoğu zaman bir "yamalı çanta(rag bag)"mevcuttur. Yamalı çanta, başka nesnelerle birleşemeyen nesneleri içerir. Yamalı çantada bulunan bu nesnelerin içerik olarak kategorisi genellikle "karışık", "diğer" ve benzeri sıfatlarla adlandırılır. Yamalı çanta kullanılması durumu şu yüzdendir. Saf bir küme içerisine heterojen bir nesnenin konması durumunda oluşacak hata payı, bu nesnenin yamalı çantaya konması durumuna göre daha fazladır. Bu yüzden yamalı çanta kullanılır. Örneğin bir C_1 kümelemesi düşünelim. Bu kümede bir nesne hariç hepsi aynı kategoride sadece bir nesne başka kategoriye ait fakat bütün nesneler yine de aynı küme altında toplanıyor. C_2 kümelemesinde ise herşey C_1 ile aynı fakat diğer nesnelere benzemeyen nesne, kendine benzemeyen nesnelerle aynı kümeye konmaktansa yamalı çantaya yani bir başka kümeye konuluyor. Yani bir başka deyişle; C_2 kümelemesinde yamalı çanta kullanıyor. Bu durumda kümelemenin kalitesini ifade eden Q değeri C_2 kümelemesinde daha yüksek bir değer içerir. Q(C_2, Cg) > Q(C_1, Cg).

Kümelemenin Kalitesini Ölçmek (Measuring Clustering Quality)

Yapı Dışı metodlar (Extrinsic Methods)

(iv) **Küçük kümelerin korunması.** Eğer az sayıda nesne içeren bir kategori daha küçük bir şekilde bölümlenirse, bu sefer bu küçük parçalar gürültülü veri haline dönüşebilir ve kümeleme işleminde farkedilemez hale gelir. Küçük kümelerin korunması kriteri küçük verilerin parçalanmasının, geniş kategorilerin parçalanmasından daha tehlikeli olduğunu vurgular. biraz aşırı bir örnek düşünelim. $\mathbf{o_1, \ldots, o_n}$ arasındaki veriler bir kategori$\mathbf{o_{n+1}, o_{n+1}}$ ise ayrı bir kategori olsun. C_1 kümelemesin üç tane kümesi olsun ve küme içerikleri, $\boldsymbol{C_1 = \{o_1, \ldots, o_n\}, C_2 = \{o_{n+1}\},}$ ve $\boldsymbol{C_3 = \{o_{n+2}\}}$ olsun. C_2 kümelemesinde de yine aynı değerler olsun fakat küme içerikleri bu sefer şöyle olsun. $\boldsymbol{C_1 = \{o_1, \ldots, o_{n-1}\}, C_2 = \{ o_n\},}$ ve$\boldsymbol{C_3 = \{o_{n+1}, o_{n+2}\}.}$ Yani bir başka deyişle C_1 küçük kategoriyi daha da küçük bir hale getirmiştir. C_2 ise büyük kategoriyi bölmüştür. Bu durumda kümeleme kalitesi $Q(C_2, C_g) > Q(C_1, C_g)$ olur.

Kümelemenin Kalitesini Ölçmek (Measuring Clustering Quality)

- **Yapı İçi Metodlar (Intrinsic Methods)**

Veri kümesinde bölge araştırması(ground truth) mümkün olmadığı zaman, kümelemenin kalitesini belirlemek için yapı içi metodları kullanmamız gerekir. Genel olarak, yapı içi metodları kümelemeyi, kümelerin ne kadar iyi bir şekilde bölündüğüne ve yoğunlaştırıldığına bakarak inceler. Bir çok yapı içi metodunun, nesneler ve veri kümeleri arasında benzerlik ölçülü avantajı vardır.

Siluet katsayısı (silhouette coefficient) bir ölçüdür. Bir veri kümesi, D, *n* tane nesne, ve farzedelim ki *D*, *k* adet kümeye bölünüyor, $C_1, \ldots, C_k$. Her $o \in D$ nesnesi için, o nesnesinin bulunduğu kümede, **o** ile diğer nesneler arasındaki ortalama uzaklığı hesaplarız ve bunu $a(o)$ ' ya eşitleriz. Benzer bir şekilde, $\beta(o)$, **o** nesnesinin bulunduğu kümeden diğer kümelere olan minimum ortalama uzaklığı hesaplar. Formal olarak, farzedelim ki $o \in C_i$ $(1 \le i \le k)$; o halde;

$$a(o) = \frac{\sum_{o' \in C_i, o \neq o'} dist(o,o')}{|C_i| - 1} \qquad (10.31)$$

ve

$$\beta(o) = min_{C_j : 1 \le j \le k, j \neq i} \left\{ \frac{\sum_{o' \in C_j} dist(o,o')}{|C_j|} \right\} \qquad (10.32)$$

o'nun siluet katsayısı şöyle tanımalanır :

Kümelemenin Kalitesini Ölçmek (Measuring Clustering Quality)

- **Yapı İçi Metodlar (Intrinsic Methods)**

o'nun siluet katsayısı şöyle tanımalanır :

$$\gamma(0) = \frac{\beta(o) - \alpha(o)}{max\{\alpha(o),\beta(o)\}} \qquad (10.33)$$

Siluet katsayısının değeri -1 ile 1 arasındadır. $a(o)$ değeri **o** nesnesinin ait olduğu kümenin yoğunluğunu temsil eder. Bu değer küçüldükçe, daha yoğun bir küme oluşur. $\beta(o)$ ise hangi **o**'nun diğer kümelere ayrıştığının derecesiyle alakalıdır. Daha büyük $\beta(o)$ değeri, **o**'nun diğer kümelerden daha da ayrık olmasını ifade eder. Ne zaman **o**'nun siluet katsayısı 1'e yaklaşır, o nesnesini içeren kümenin yoğun ve o nesnesinin diğer kümelerden uzak olduğuna işarettir ki bu istenen durumdur. Aksi durum ise, yani o'nun siluet katsayısının negatif olması, o'nun bir kümeye o kümenin içindekilerden daha yakın olma durumudur ki bu çoğu durum da istenmeyen ve kaçınılması gereken bir durumdur.

Kümeleme de kümelerin durumunu ölçmek için, kümede ki her nesne için ortalama siluet katsayı hesaplanabilir. Kümelemenin kalitesini ölçmek için, veri kümesindeki bütün nesnelerin ortalama siluet katsayısını kullanabiliriz. Siluet katsayısı ve başka iç yapı ölçümleri dirsek metodunda (elbow method) da kullanılabilir.