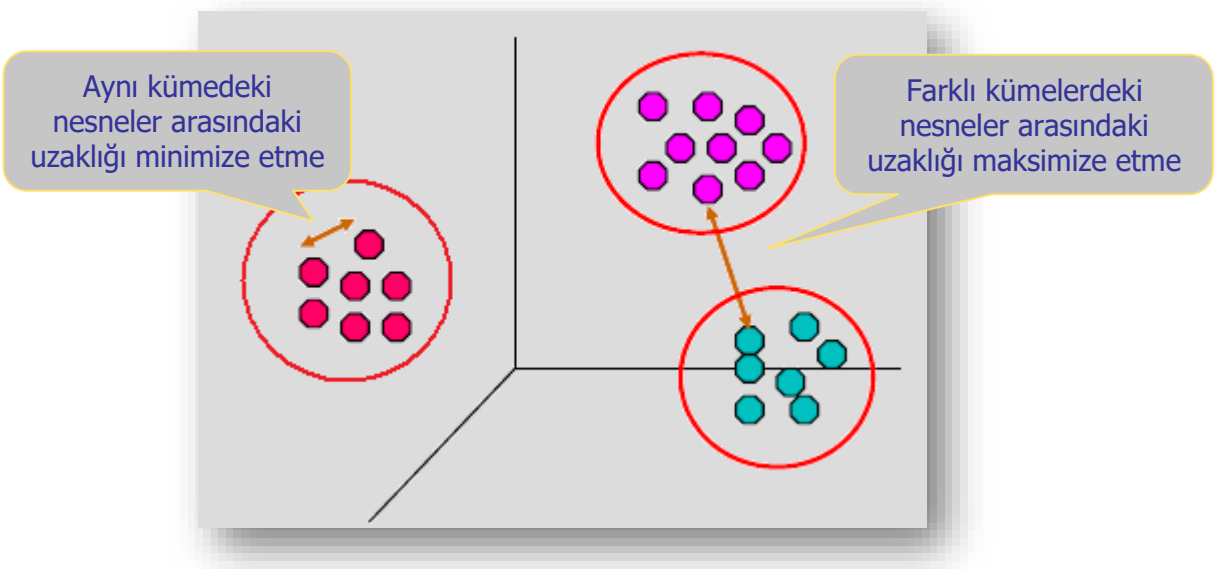
Kümeleme

- Özelliklerine göre veriler arasındaki benzerliklerin bulunması ve benzer veri nesnelerinin kümeler halinde gruplanması
- Küme: Birbirine benzeyen nesnelerden oluşan grup
  - Aynı kümedeki nesneler birbirine daha çok benzer
  - Farklı kümelerdeki nesneler birbirine daha az benzer



Kümeleme

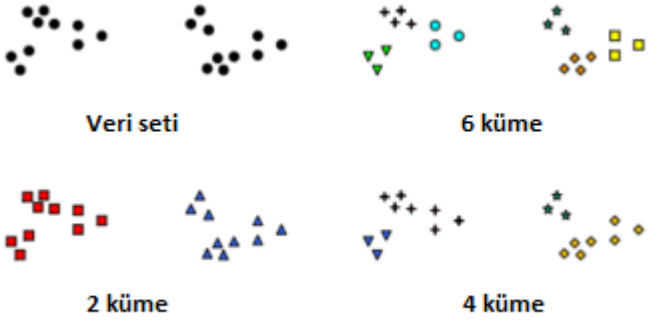
- Sınıflandırma işleminde sınıflar önceden belirli iken kümelemede sınıflar belli değildir.
- Denetimsiz öğrenme
  - Hangi nesnenin hangi sınıfa ait olduğu ve sınıf sayısı belli değil
- Gruplandırma eldeki verilerin benzerliğine göre yapılır.
- Uygulamaları
  - Veri dağılımına ilişkin bilgi edinmek için bağımsız bir araç olarak
  - Diğer veri madenciliği uygulamaları için bir ön işleme adımı olarak

Kümeleme Uygulamaları

- Görüntü işleme
- Ekonomi
- Biyoloji
- Doküman kümeleme
- Kullanıcı kümeleme
- Kullanıcı davranışlarını kümeleme
- Pazarlama
- Şehir planlaması
- Diğer veri madenciliği uygulamaları için bir ön işleme adımı olarak
  - Veri azaltma – küme içindeki nesnelerin temsil edilmesi için küme merkezlerinin kullanılması
  - Sıkıştırma

İyi Kümeleme

- İyi bir kümeleme yöntemi yüksek kaliteli kümeler üretir.
  - Küme içi benzerlik fazla
  - Kümeler arası benzerlik düşük
- Bir kümeleme yönteminin kalitesi
  - kullanılan benzerlik ölçütüne
  - bu ölçütün uygulanmasına
  - gizli örüntülerin bir kısmını veya tamamını keşfetme yeteneğine bağlıdır.

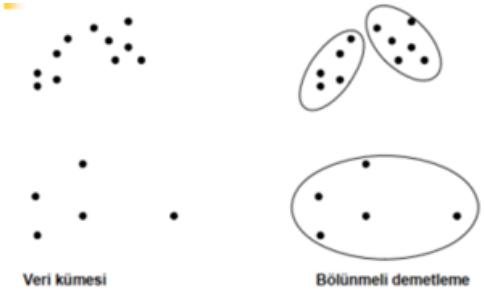


Kümeleme Yöntemleri

- **Bölünmeli yöntemler:**
  - Veriyi bölerek, her grubu belirlenmiş bir kritere göre değerlendirir.
  - k-means, k-medoids, CLARANS
- **Hiyerarşik yöntemler:**
  - Veri kümelerini (ya da nesneleri) önceden belirlenmiş bir kritere göre hiyerarşik olarak ayırır.
  - Diana, Agnes, BIRCH, CAMELEON
- **Yoğunluk tabanlı yöntemler:**
  - Nesnelerin yoğunluğuna göre kümeleri oluşturur.
  - DBSCAN, OPTICS, DenClue
- **Model tabanlı yöntemler:**
  - Her kümenin bir modele uyduğu varsayılır. Amaç bu modellere uyan verileri gruplamaktır.
  - EM, SOM, COBWEB

Bölünmeli Yöntemler

- Amaç: **n** nesneden oluşan bir veri kümesini **(D), k (k≤n)** kümeye ayırmak
  - her demette en az bir nesne bulunmalı
  - her nesne sadece bir demette bulunmalı
- Yöntem: Kümeleme kriterini en büyütücek şekilde **D** veri kümesi **k** gruba ayırma
  - Global çözüm: Mümkün olan tüm gruplamaları yaparak en iyisini seçme
  - Sezgisel çözüm: k-means ve k-medoids
    - k-means : Her demet kendi merkezi ile temsil edilir.
    - k-medoids veya PAM : Her demet, demette bulunan bir nesne ile temsil edilir.



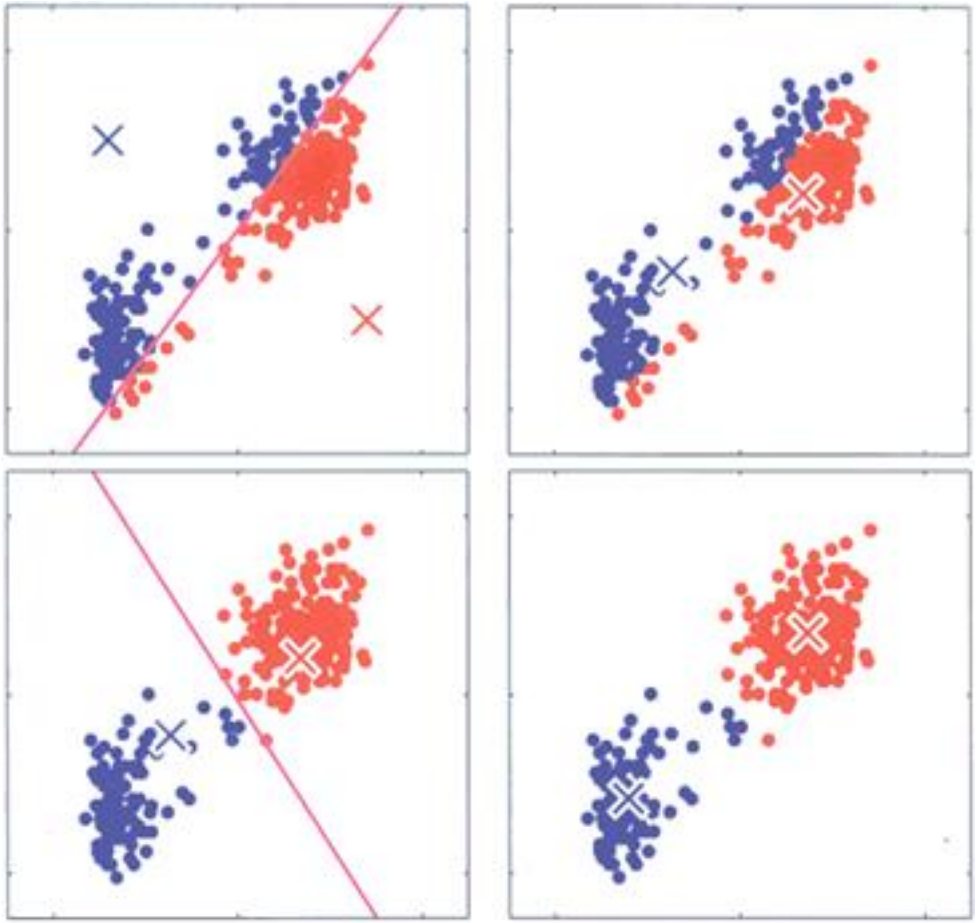
K-means kümeleme

- Bilinen bir k değeri için k-means kümeleme algoritmasının 4 aşaması vardır:

1. Veri kümesi k altkümeye ayrılır (her küme bir altküme)
2. Her kümenin ortalaması hesaplanır:merkez nokta (kümedeki nesnelerin niteliklerinin ortalaması)
3. Her nesne en yakın merkez noktanın olduğu kümeye dahil edilir.
4. Nesnelerin kümelenmesinde değişiklik olmayana kadar adım 2'ye geri dönülür.

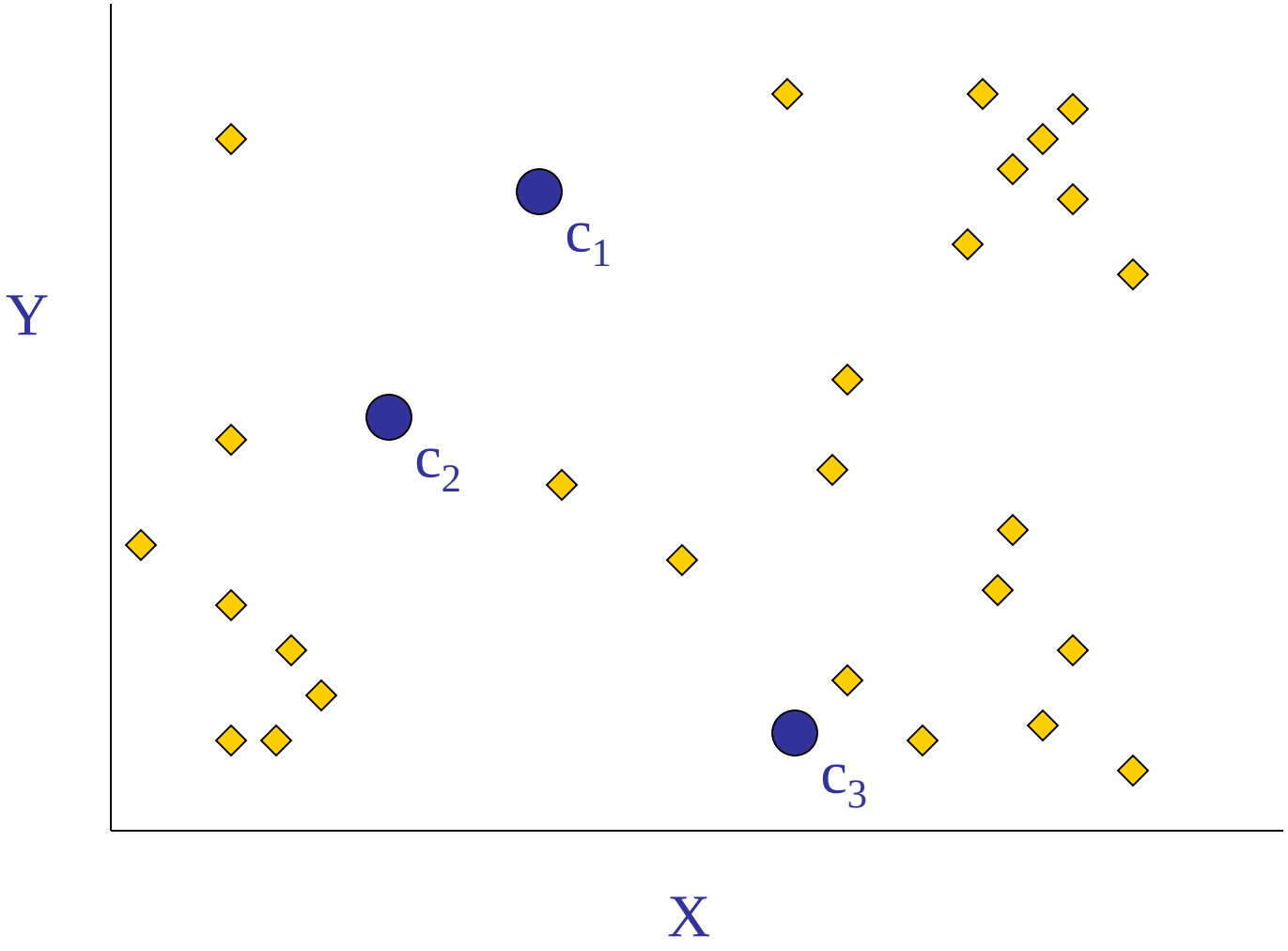
K-means kümeleme

- Veri içerisinden rasgele seçilen K adet noktaya küme merkezi gözüyle bakılır. Tüm veri noktaları bu küme merkezlerine uzaklıklarına göre gruplanır. Her gruplamadan sonra küme merkezleri tekrar hesaplanır.



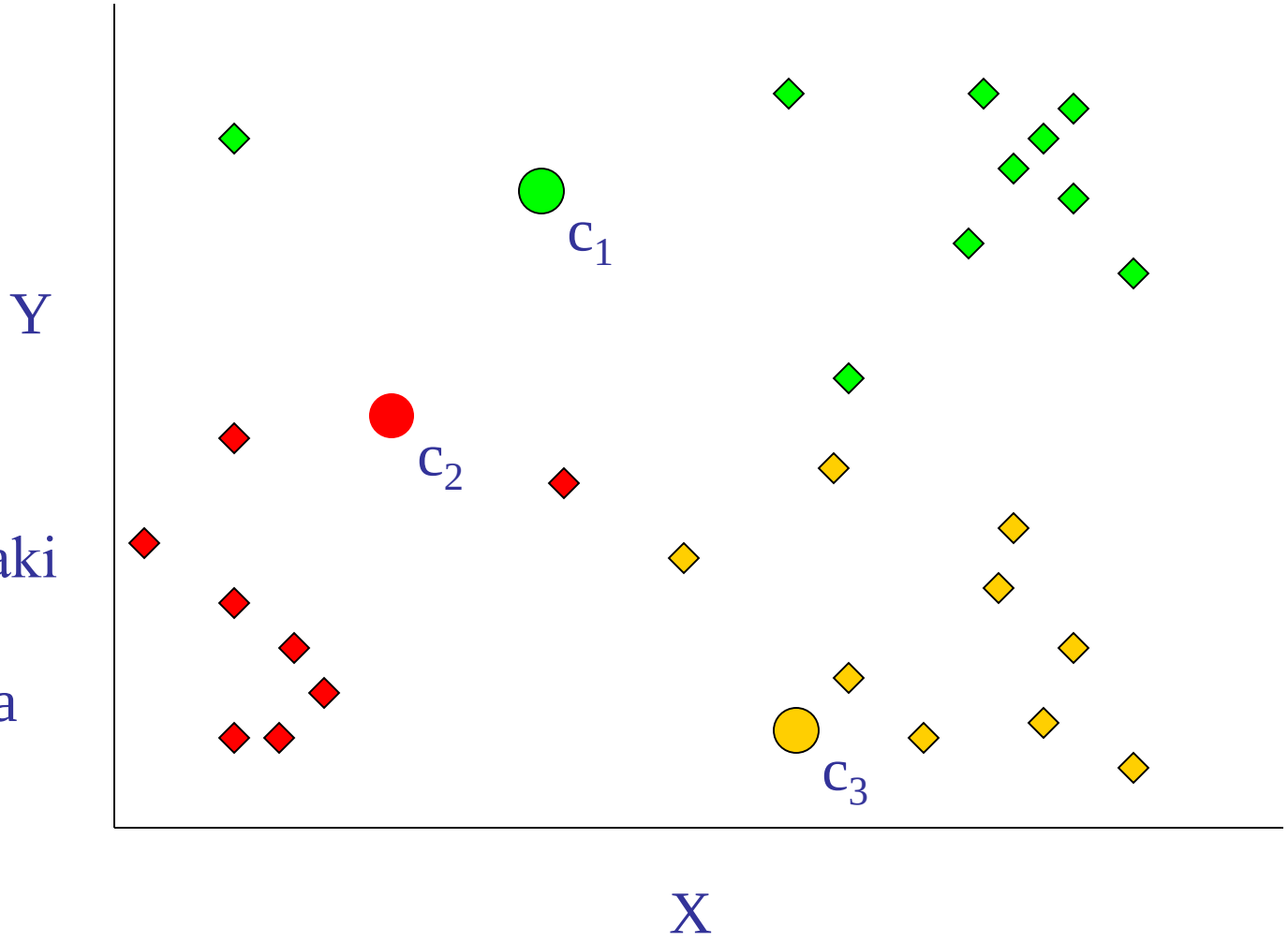
K-means kümeleme - Örnek

Rasgele 3 küme merkezi ata



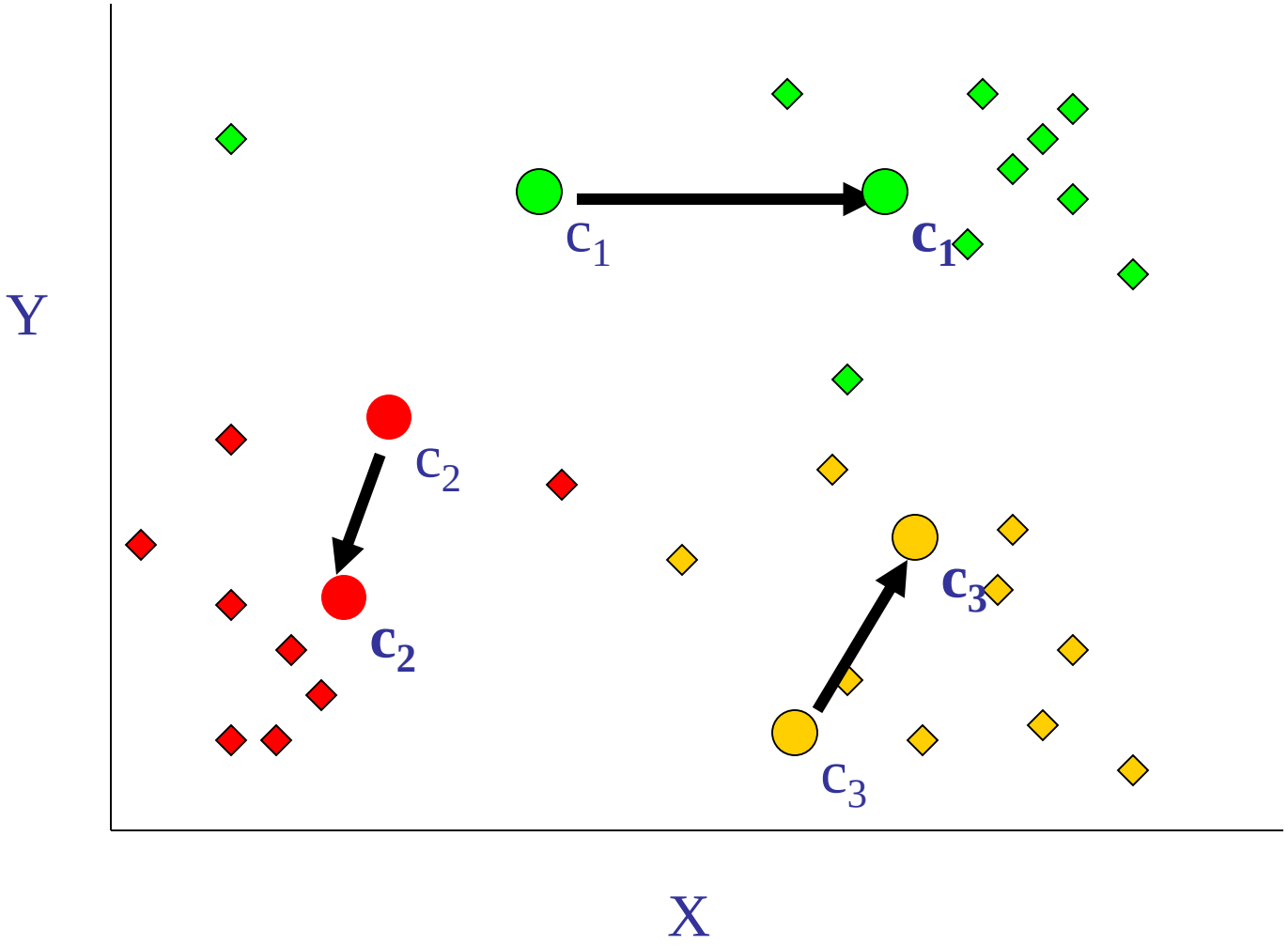
K-means kümeleme - Örnek

Her örneği en yakınındaki merkezin kümesine ata



K-means kümeleme - Örnek

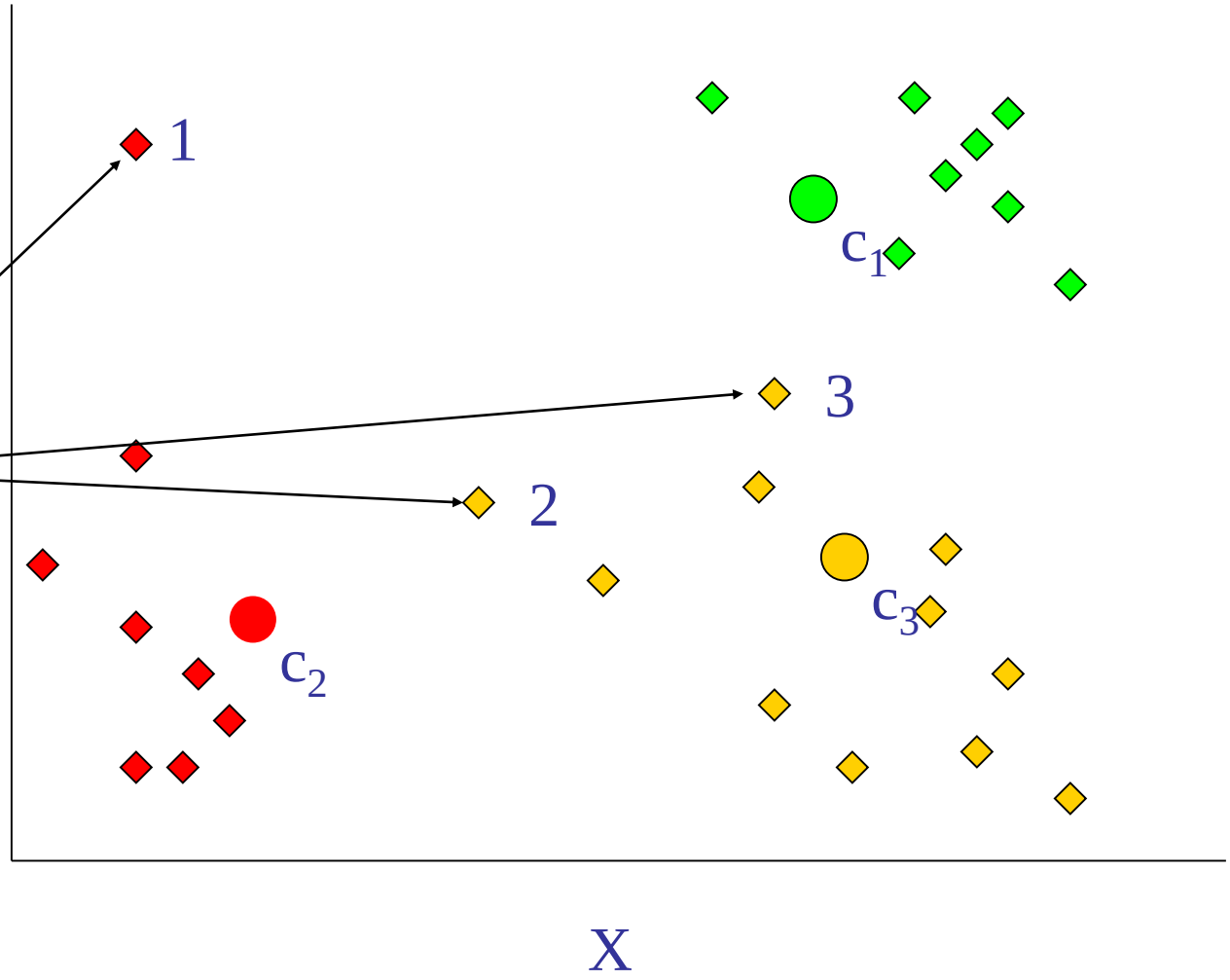
Merkezleri kendi kümelerinin merkezine götür.



K-means kümeleme - Örnek

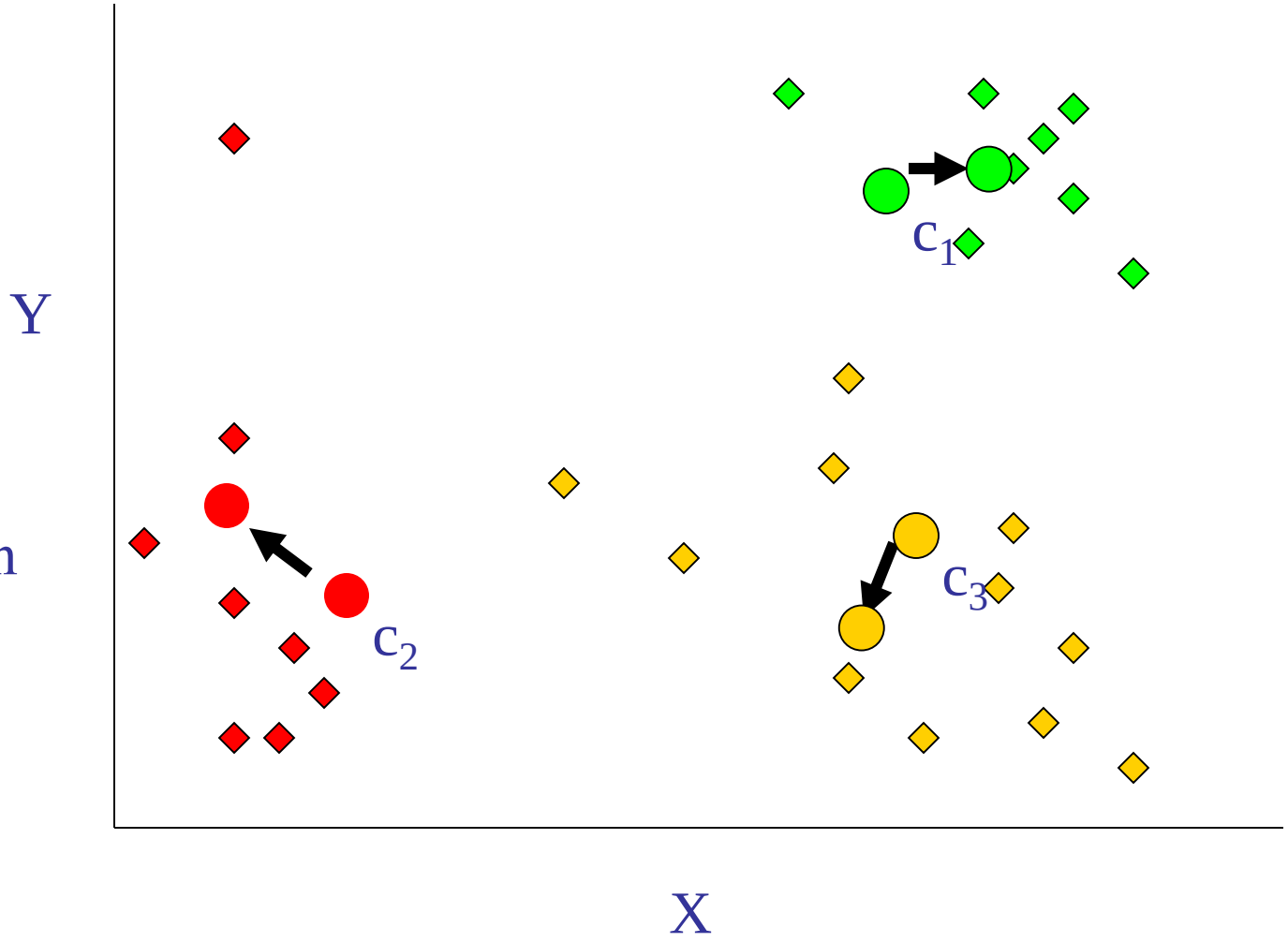
Her örneği yeniden en yakınındaki merkezin kümesine ata.

Q: Hangi örneklerin kümesi değişti?

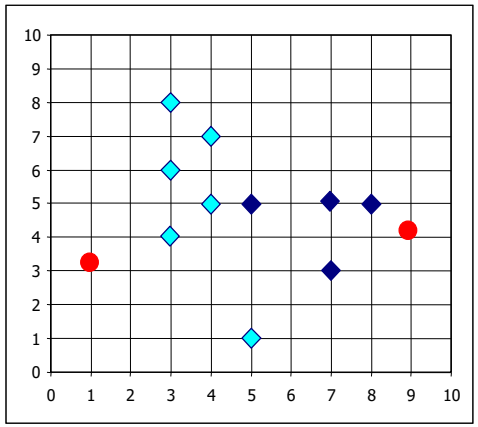


K-means kümeleme - Örnek

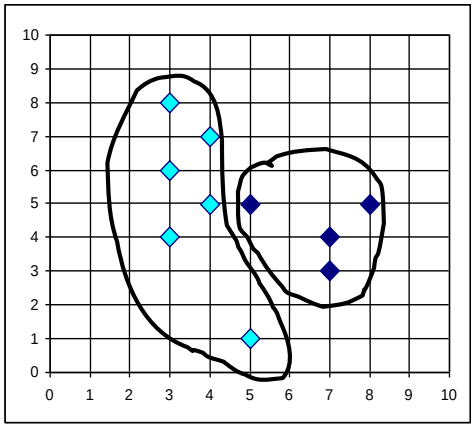
Merkezleri kendi kümelerinin merkezine götür.



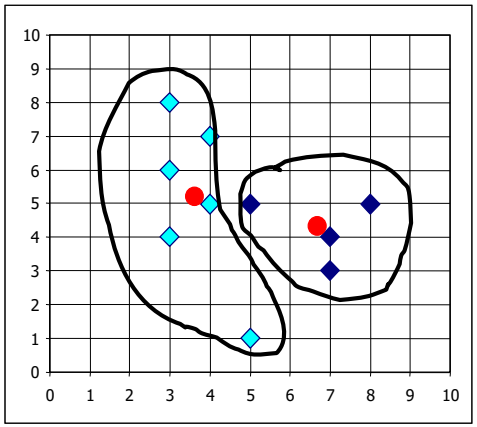
K-means kümeleme - Örnek



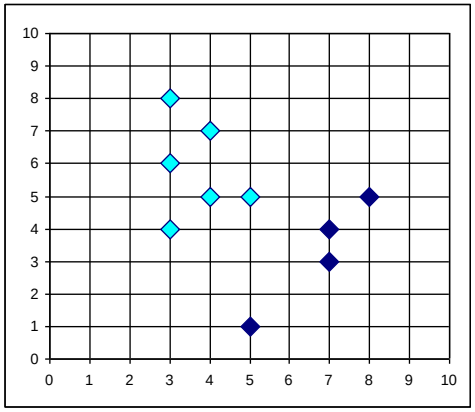
Her nesneyi en benzer merkeze ata



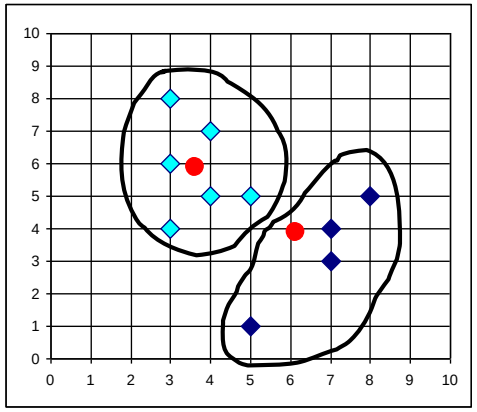
Küme ortalamalarını güncelle



Tekrar ata



Küme ortalamalarını güncelle



Tekrar ata

K=2

Başlangıç küme merkezi olarak gelişigüzel K nesne seç

K-means kümeleme

- Yaygın olarak kullanılan yöntem hataların karelerinin toplamı (Sum of Squared Error SSE)
  - Nesnelerin bulundukları kümenin merkez noktalarına olan uzaklıklarının karelerinin toplamı

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2 (m_i, x)$$

x : C_i demetinde bulunan bir nesne

m_i : C_i demetinin merkez noktası

- Hataların karelerinin toplamını azaltmak için k demet sayısı artırılabilir.
  - Küçük k ile iyi bir demetleme, büyük k ile kötü bir demetlemeden daha az SSE değerine sahip olabilir.
- Başlangıç için farklı merkez noktaları seçerek farklı demetlemeler oluşturulur.
- En az SSE değerini sahip olan demetleme seçilir.

K-means kümeleme - Örnek

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

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

K-means kümeleme - Örnek

- Kümelerin sayısına başlangıçta k=2 biçiminde karar veriyoruz. Başlangıçta rasgele olarak aşağıdaki iki kümeyi belirliyoruz.

$$C_1=\{X_1, X_2, X_4\}$$
$$C_2=\{X_3, X_5\}$$

Gözlemler	Değişken 1	Değişken 2	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

K-means kümeleme - Örnek

İki kümenin merkezleri şu şekilde hesaplanır:

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

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

$\longrightarrow$

Küme içi değişmeler şu şekilde hesaplanır:

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

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

Toplam karesel hata aşağıdaki gibi hesaplanır:

$$E^2 = e_1^2 + e_2^2 = 26.67 + 42.50 = 69.17$$

K-means kümeleme - Örnek

Gözlemlerin M_1 ve M_2 merkezlerinden olan uzaklıkların minimum olması istendiğinden aşağıdaki hesaplamalar yapılır:

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

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

$d(M_1,X_1) < d(M_2,X_1)$ olduğundan M_1 merkezinin X_1 gözlem değerine daha yakın olduğu anlaşılır. O halde $X_1 \in C_1$ olarak kabul edilir.

K-means kümeleme - Örnek

Tüm gözlem değerleri için hesaplamalar yapılarak aşağıdaki tablo elde edilir:

Gözlemler	M_1den uzaklık	M_2den 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

$$C_1 = \{X_1, X_2, X_3\}$$
$$C_2 = \{X_4, X_5\}$$

K-means kümeleme - Örnek

İki kümenin merkezleri tekrar hesaplanır:

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

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

→

Küme içi değişmeler tekrar hesaplanır:

$$e_1^2 = \left[(4-5)^2 + (2-2.33)^2\right] + \left[(6-5)^2 + (4-2.33)^2\right] + \left[(5-2.33)^2 + (1-2.33)^2\right] = 9.33$$

$$e_2^2 = \left[(10-10.5)^2 + (6-7)^2\right] + \left[(11-10.5)^2 + (8-7)^2\right] = 2.50$$

Toplam karesel hata aşağıdaki gibi hesaplanır:

$$E^2 = e_1^2 + e_2^2 = 9.33 + 2.5 = 11.83$$

Bu değerin bir önceki iterasyonda elde edilen E^2=69.17 değerinden daha küçük olduğu anlaşılır.

K-means kümeleme - Örnek

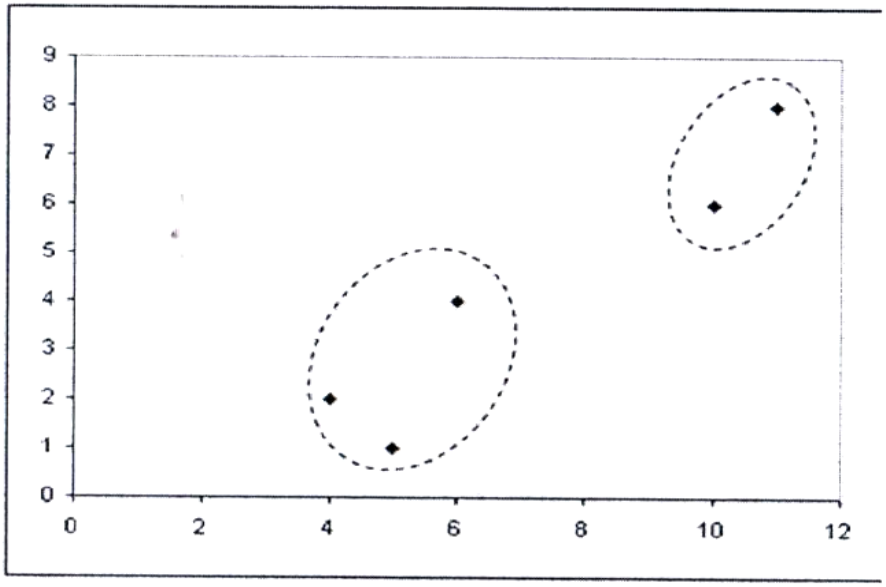
M_1 ve M_2 merkezinden gözlem değerlerine olan uzaklıklar hesaplandığında aşağıdaki tablo elde edilir.

Gözlemler	M_1den uzaklık	M_2den 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

$$C_1 = \{X_1, X_2, X_3\}$$
$$C_2 = \{X_4, X_5\}$$

K-means kümeleme - Örnek

Kümelerde önceki adıma göre herhangi bir değişme olmadığına göre iterasyona burada son verilir. Elde edilen kümeler aşağıdaki şekilde gösterilmiştir.



Şekil-6.18. Sonuç olarak elde edilen kümeler

K-means kümeleme

- Demet sayısının belirlenmesi gerekir.
- Başlangıçta demet merkezleri rasgele belirlenir.
  - Her uygulamada farklı demetler oluşabilir.
- Uzaklık ve benzerlik Öklid uzaklığı, kosinüs benzerliği gibi yöntemlerle ölçülebilir.
- Az sayıda tekrarda demetler oluşur.
  - Yakınsama koşulu çoğunlukla az sayıda nesnenin demet değiştirmesi şekline dönüştürülür.
- Karmaşıklığı:
  - Yer karmaşıklığı - O((n+k) d)
  - Zaman karmaşıklığı - O(ktnd)

k: demet sayısı, t: tekrar sayısı, n: nesne sayısı, d: nitelik sayısı

K-means kümeleme

- Gerçeklemesi kolay
- Karmaşıklığı diğer demetleme yöntemlerine göre az
- K-means algoritması bazı durumlarda iyi sonuç vermeyebilir
  - Veri grupları farklı boyutlarda ise
  - Veri gruplarının yoğunlukları farklı ise
  - Veri gruplarının şekli küresel değilse
  - Veri içinde aykırılıklar varsa

K-medoids kümeleme

- K-medoids algoritmasının temeli, verinin çeşitli yapısal özelliklerini temsil eden k tane temsilci nesneyi bulma esasına dayanır.
- Bir grup nesneyi k tane kümeye bölerken esas amaç, birbirine çok benzeyen nesnelerin bir arada olduğu ve farklı kümelerdeki nesnelerin mümkün olduğunca birbirinden benzersiz olduğu kümeleri bulmaktır.
- Amacın k tane nesneyi bulmak olmasından dolayı, K-medoids metodu olarak adlandırılmaktadır.

K-medoids kümeleme

- Temsilci nesne diğer nesnelere olan ortalama uzaklığı minimum yapan kümenin en merkezi nesnesidir.
- Bu nedenle, bu bölünme metodu her bir nesne ve onun referans noktası arasındaki benzersizliklerin (uzaklıkların) toplamını küçültme mantığı esas alınarak uygulanır.
- k adet temsilci nesne tespit edildikten sonra her bir nesne en yakın olduğu temsilciye atanarak k tane küme oluşturulur.
- Sonraki adımlarda her bir temsilci nesne temsilci olmayan nesne ile değiştirilerek kümelemenin kalitesi yükseltilinceye kadar ötelenir.
- Bu kalite nesne ile ait olduğu kümenin temsilci nesnesi arasındaki ortalama benzersizliğe göre değişir.

K-medoids kümeleme

- PAM (Partitioning Araound Medoids)
  - Başlangıçta k adet nesne kümeleri temsil etmek üzere rasgele seçilir x_{ik}
  - Kalan nesneler en yakın merkez nesnenin bulunduğu kümeye dahil edilir
  - Merkez nesne olmayan rasgele bir nesne seçilir x_{rk}
  - x_{rk}merkez nesne olursa toplam karesel hatanın ne kadar değiştiği bulunur

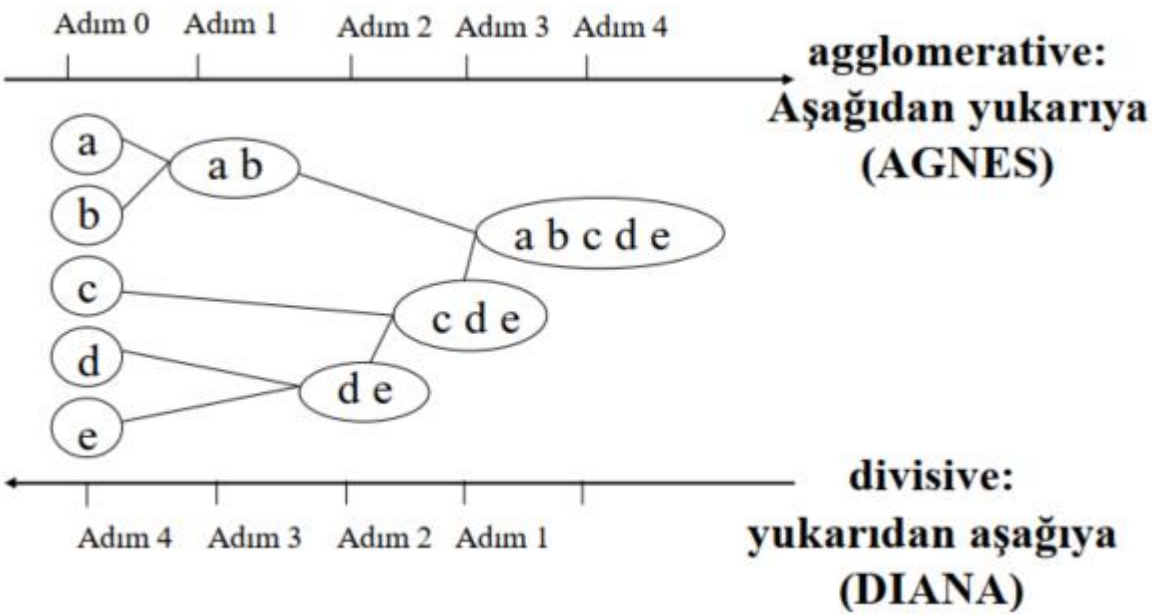
$$TC_{ik} = \sum_{J=1}^{n_k} (x_{ik} - x_{jk})^2 - \sum_{J=1}^{n_k} (x_{rk} - x_{jk})^2$$

n_k: k kümesi içindeki nesne sayısı
x_{jk}: k kümesi içindeki j. nesne

  - TC_{ik}<0 ise O_{rk} merkez nesne olarak atanır.
  - Kümelerde değişiklik oluşmayana kadar 3. adıma geri gidilir.
- Küçük veri kümeleri için iyi sonuç verebilir, ancak büyük veri kümeleri için uygun değil

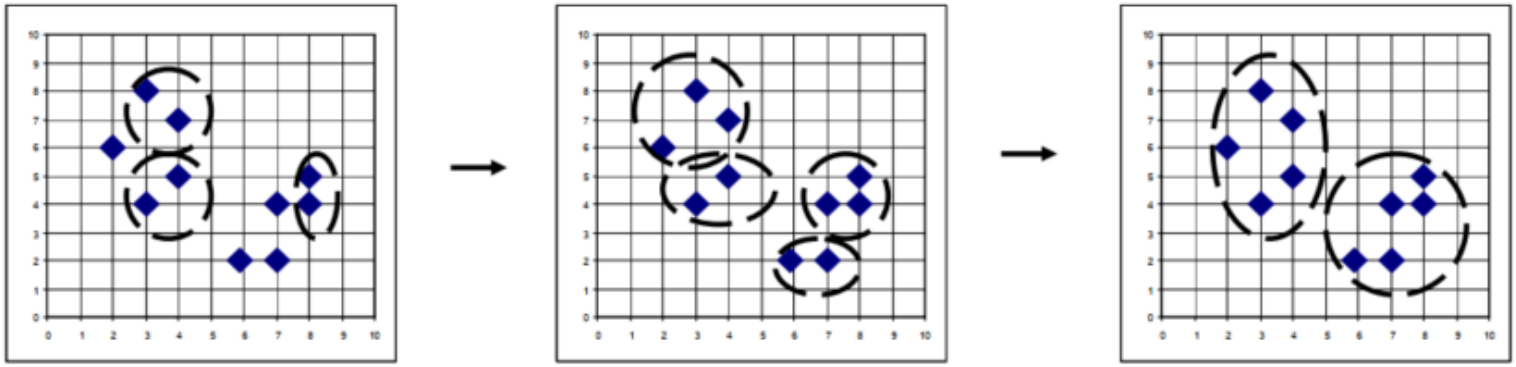
Hiyerarşik kümeleme

- Demet sayısının belirlenmesine gerek yok
  - Sonlanma kriteri belirlenmesi gerekiyor



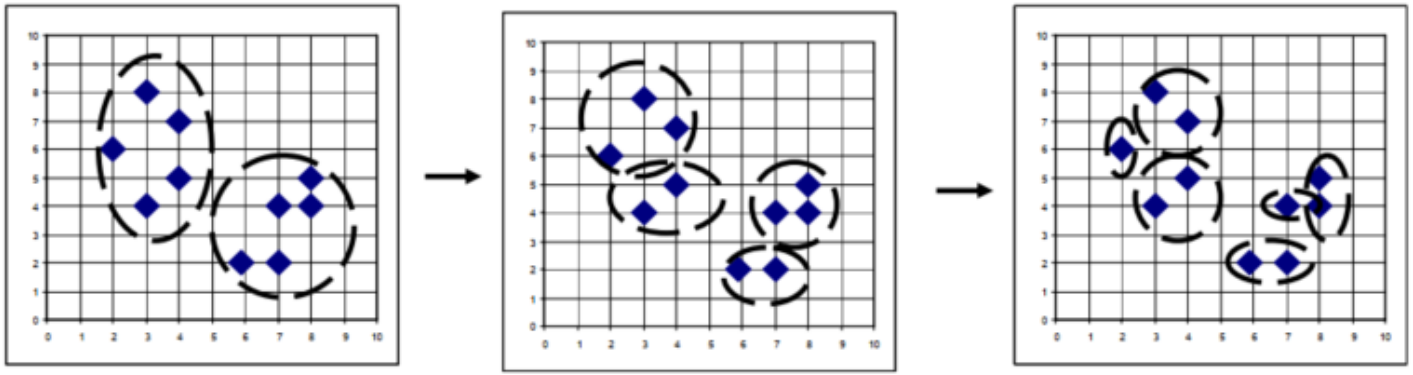
Hiyerarşik kümeleme

- AGNES (AGglomerative NESting):
  - Birinci adımda her nesne bir küme oluşturur.
  - Aralarında en az uzaklık bulunan kümeler her adımda birleştirilir.
  - Bütün nesneler tek bir küme içinde kalana kadar ya da istenen sayıda küme elde edene kadar birleştirme işlemi devam eder.



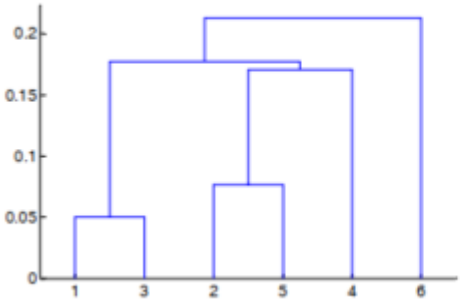
Hiyerarşik kümeleme

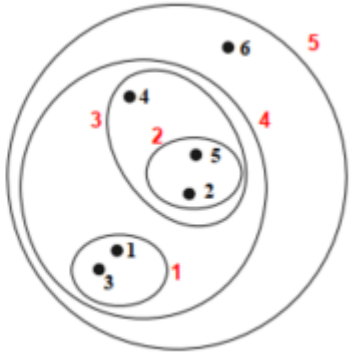
- DIANA (DIvisive ANAlysis):
  - AGNES'in yaptığı işlemlerin tersini yapar.
  - En sonunda her nesne bir küme oluşturur.
  - Her nesne ayrı bir küme oluşturana ya da istenilen küme sayısı elde edene kadar ayrılma işlemi devam eder.



Hiyerarşik kümeleme

- Dendogram: Kümeler hiyerarşik olarak ağaç yapısı şeklinde görüntülenebilir.
- Ara düğümler çocuk düğümlerdeki kümelerin birleşmesiyle elde edilir.
  - Kök: bütün nesnelerden oluşan tek küme
  - Yapraklar: bir nesneden oluşan kümeler
- Dendogram istenen seviyede kesilerek kümeler elde edilir.

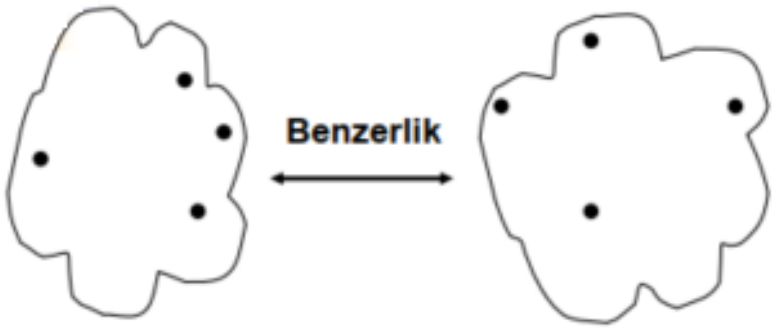




Aşağıdan yukarıya kümeleme

- Algoritma
  1. Uzaklık matrisini hesapla
  2. Her nesne bir küme
  3. Tekrarla
  4. En yakın iki kümeyi birleştir
  5. Uzaklık matrisini yeniden hesapla
  6. Sonlanma: Tek bir küme kalana kadar
- Uzaklık matrisini hesaplarken farklı yöntemler farklı kümeleme sonuçlarına neden olurlar.

Demetler arası uzaklık

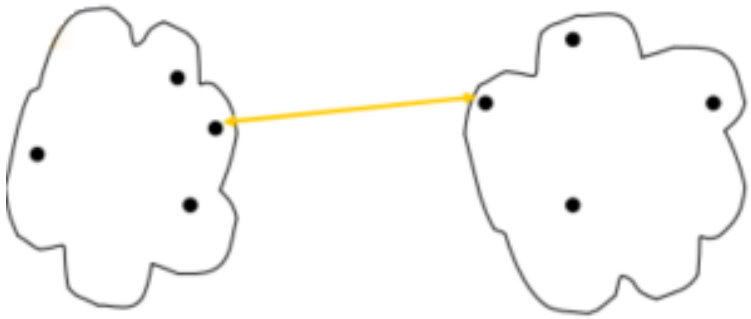


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Uzaklık Matrisi

- MIN (Tek bağ)
- MAX (Tam bağ)
- Ortalama
- Merkezler arası uzaklık

Demetler arası uzaklık

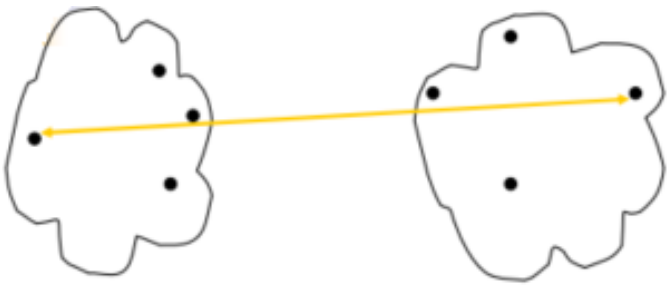


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Uzaklık Matrisi

- MIN (Tek bağ)
- MAX (Tam bağ)
- Ortalama
- Merkezler arası uzaklık

Demetler arası uzaklık

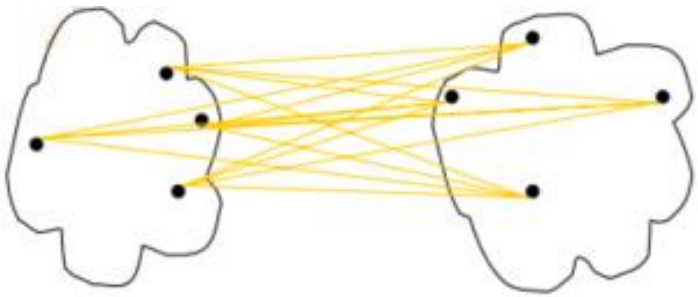


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Uzaklık Matrisi

- MIN (Tek bağ)
- MAX (Tam bağ)
- Ortalama
- Merkezler arası uzaklık

Demetler arası uzaklık

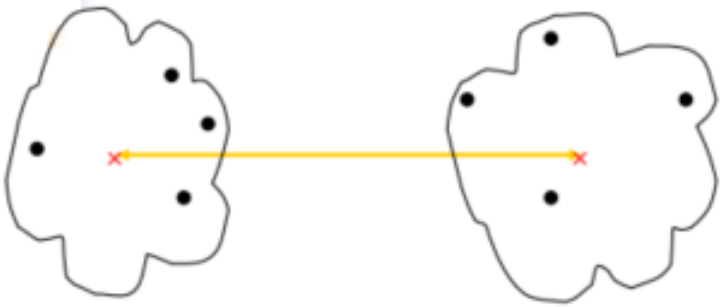


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Uzaklık Matrisi

- MIN (Tek bağ)
- MAX (Tam bağ)
- Ortalama
- Merkezler arası uzaklık

Demetler arası uzaklık

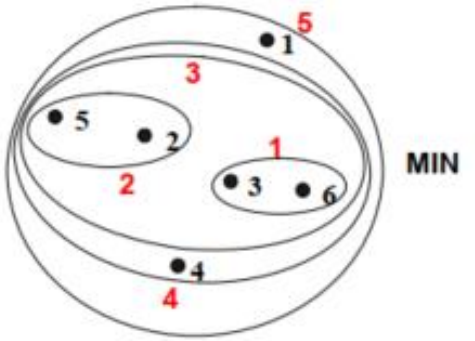


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Uzaklık Matrisi

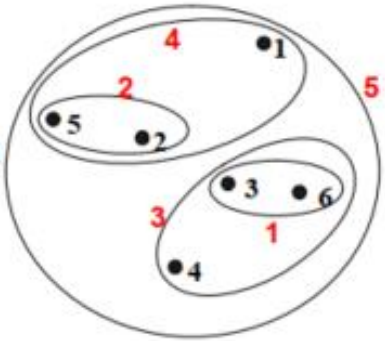
- MIN (Tek bağ)
- MAX (Tam bağ)
- Ortalama
- Merkezler arası uzaklık

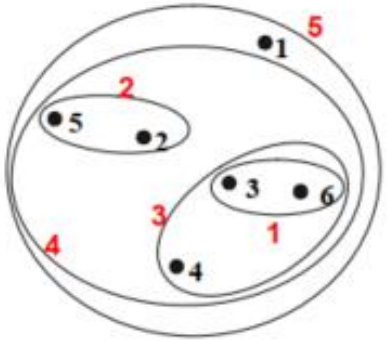
Farklı Uzaklık Yöntemlerinin Etkisi



MIN

MAX





Ortalama

Hiyerarşik kümeleme

- Demetleme kriteri yok
- Demet sayılarının belirlenmesine gerek yok
- Aykırılıklardan ve hatalı verilerden etkilenir
- Farklı boyuttaki demetleri oluşturmak problemli olabilir
- Yer karmaşıklığı – $O(n^2)$
- Zaman karmaşıklığı – $O(n^2 logn)$

n : nesne sayısı

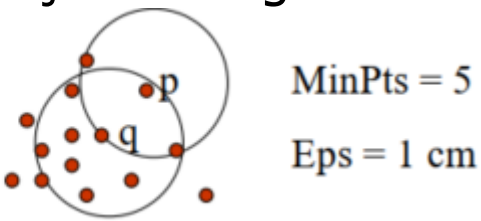
Yoğunluk tabanlı yöntemler

- Demetleme nesnelerin yoğunluğuna göre yapılır.
- Başlıca özellikleri:
  - Rasgele şekillerde demetler üretilebilir.
  - Aykırı nesnelerden etkilenmez.
  - Algoritmanın son bulması için yoğunluk parametresinin verilmesi gerekir.

DBSCAN

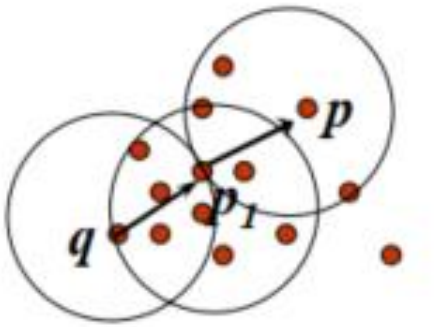
- İki parametre:
  - Eps: En büyük komşuluk yarıçapı
  - MinPts: Eps yarıçaplı komşuluk bölgesinde bulunan en az nesne sayısı
- $N_{eps}(p)$: {q∈D | d(p,q)≤Eps}
- Doğrudan erişilebilir nesne: Eps ve MinPts koşulları altında bir q nesnesinin doğrudan erişilebilir bir p nesnesi şu şartlar sağlar:
- $p \in N_{eps}(q)$
- q nesnesinin çekirdek nesne koşulunu sağlaması

$N_{eps}(q) \geq MinPts$

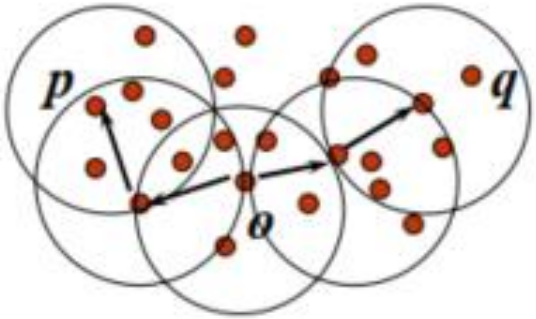


DBSCAN

- Erişilebilir nesne:
  - Eps ve MinPts koşulları altında q nesnesinin erişilebilir bir p nesnesi olması için:
    - p1,p2,...,pn nesne zinciri olması,
    - p1=q, pn=p,
    - p_i nesnesinin doğrudan erişilebilir nesnesi:p_{i+1}



- Yoğunluk bağlantılı Nesne:
  - Eps ve MinPts koşulları altında q nesnesinin yoğunluk bağlantılı nesnesi p şu koşulları sağlar:
    - p ve q nesneleri Eps ve MinPts koşulları altında bir o nesnesinin erişilebilir nesnesi



DBSCAN

- Veri tabanındaki her nesnenin Eps yarıçaplı komşuluk bölgesi araştırılır.
- Bu bölgede MinPts'den daha fazla nesne bulunan p nesnesi çekirdek nesne olacak şekilde kümeler oluşturulur.
- Çekirdek nesnelerin doğrudan erişilebilir nesneleri bulunur.
- Yoğunluk bağlantılı kümeler birleştirilir.
- Hiçbir yeni nesne bir kümeye eklenmezse işlem sona erer.
- Yer karmaşıklığı – O(n)
- Zaman karmaşıklığı – O(nlogn)

n: nesne sayısı

Model tabanlı yöntemler

- Veri kümesi için öngörülen matematiksel model en uygun hale getiriliyor.
- Verinin genel olarak belli olasılık dağılımlarının karışımından geldiği kabul edilir.
- Model tabanlı demetleme yöntemi
  - Modelin yapısının belirlenmesi
  - Modelin parametrelerinin belirlenmesi

Model tabanlı yöntemler

- İstatistiksel yaklaşım:
  - K nesneden oluşan bir veri kümesi $D=\{x_1, x_2, \ldots, x_K\}$
  - her $\mathbf{x}_i$ $(i \in [1,\ldots K])$ nesnesi Θ parametre kümesiyle tanımlanan bir olasılık dağılımından oluşturulur.
  - Olasılık dağılımının, $c_j \in \mathbf{C}=\{c_1, c_2, \ldots, c_G\}$ şeklinde G adet bileşeni vardır.
  - Her Θ_g, $g \in [1, \ldots, G]$ parametre kümesi g bileşeninin olasılık dağılımını belirleyen, Θ kümesinin ayrışık bir alt kümesidir.
  - Herhangi bir $\mathbf{x}_i$ nesnesi öncelikle, $p(c_g/\Theta) = \tau_g$ ($\Sigma_G \tau_g=1$ olacak şekilde) bileşen katsayısına (ya da bileşenin seçilme olasılığına) göre bir bileşene atanır.
  - Bu bileşen $p(\mathbf{x}_i/c_g; \Theta_g)$ olasılık dağılımına göre $\mathbf{x}_i$ değişkenini oluşturur.
  - Böylece bir $\mathbf{x}_i$ nesnesinin bu model için olasılığı bütün bileşenlerin olasılıklarının toplamıyla ifade edilebilir:

$$p(x_i \mid \Theta) = \sum_{g=1}^{G} p(c_g \mid \Theta) p(\mathbf{x}_i \mid c_g; \Theta_g)$$

$$p(x_i \mid \Theta) = \sum_{g=1}^{G} \tau_g p(\mathbf{x}_i \mid c_g; \Theta_g)$$

Model tabanlı yöntemler

- Model parametrelerinin belirlenmesi
  - Maximum Likelihood (ML) yaklaşımı

$$\ell_{ML}(\Theta_1,\ldots,\Theta_G;\tau_1,\ldots,\tau_G \mid D) = \prod_{i=1}^{K}\sum_{g=1}^{G}\tau_g p(x_i \mid c_g,\Theta_g)$$

  - Maximum Aposteriori (MAP) yaklaşımı

$$\ell_{MAP}(\Theta_1,\ldots,\Theta_G;\tau_1,\ldots,\tau_G \mid D) = \prod_{i=1}^{K}\sum_{g=1}^{G}\frac{\tau_g p(x_i \mid c_g,\Theta_g)p(\Theta)}{p(D)}$$

  - Uygulamada her ikisinin logaritması

$$L(\Theta_1,\ldots,\Theta_G;\tau_1,\ldots,\tau_G \mid D) = \sum_{i=1}^{K}\ln\sum_{g=1}^{G}\left(\tau_g p(x_i \mid c_g,\Theta_g)\right)$$

$$L(\Theta_1,\ldots,\Theta_G;\tau_1,\ldots,\tau_G \mid D) = \sum_{i=1}^{K}\ln\sum_{g=1}^{G}\left(\tau_g p(x_i \mid c_g,\Theta_g)\right) + \ln p(\Theta)$$

EM algoritması

- Veri kümesi: $D=\{x_1,x_2,\ldots,x_K\}$
- Gizli değişkenler $H=\{z_1,z_2,\ldots,z_K\}$ (her nesnenin hangi demete dahil olduğu bilgisi)
- Verinin eksik olduğu durumda, tam verinin beklenen değeri hesaplanır:

$$Q(\Theta,\Theta') = E[L_c(D,H\,|\,\Theta)\,|\,D,\Theta')$$
$$= \sum_{i=1}^{K}\sum_{g=1}^{G} p(c_g\,|\,x_i)[\ln p(x_i\,|\,c_g) + \ln \tau_g]$$

- EM Algoritmasının adımları:
  - Θ' için başlangıç değerleri atama
  - (E) Expectation: $Q(\Theta|\ \Theta')$ hesaplanması
  - (M) Maximization: argmax $Q(\Theta|\ \Theta')$