Veri Madenciliği

Bölüm 7. Kümeleme 1

Doç. Dr. Suat Özdemir http://ceng.gazi.edu.tr/~ozdemir

Demetleme

- Demetleme işlemleri
  - Tanımı
  - Uygulamalar
- Demetleme yöntemleri
  - Bölünmeli
  - Hiyeraşik
  - Yoğunluk tabanlı
  - Model tabanlı

Gözetimli & Gözetimsiz Öğrenme

- Predictive Data Mining vs. Descriptive Data Mining
- Gözetimli (Supervised) öğrenme= sınıflandırma (clasification)
  - Öğrenme kümesindeki sınıfların sayısı ve hangi nesnenin hangi sınıfta olduğu biliniyor.
- Gözetimsiz (Unsupervised) öğrenme = **demetleme (clustering)**
  - Öğrenme kümesinde hangi nesnenin hangi sınıfta olduğu bilinmiyor. Genelde sınıf sayısı da bilinmiyor.

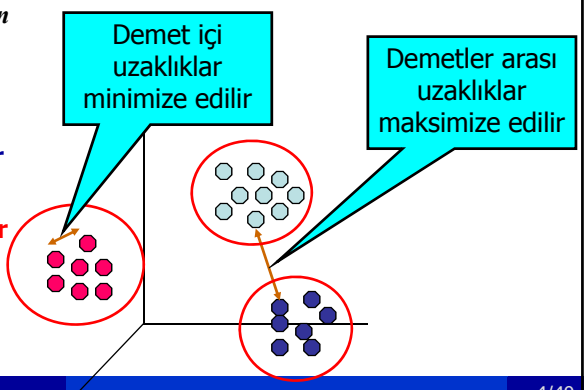
Demetleme nedir?

- Nesneleri demetlere (gruplara) ayırma
  - Karakteristiklerden yararlanarak veri içindeki benzerlikleri bulma ve benzer verileri demetler içinde gruplama

Demet/Küme: ***birbirine benzeyen nesnelerden oluşan grup***

Aynı demetteki nesneler birbirine daha çok benzer

Farklı demetlerdeki nesneler birbirine daha az benzer



Demet/Küme nedir?

Bu veride kaç demet var?

6 demet

2 demet

4 demet

Demetleme uygulama alanları

- Genel uygulama alanları:
  - verinin dağılımını anlama
  - başka veri madenciliği uygulamaları için ön hazırlık- *Veri azaltma – demet içindeki nesnelerin temsil edilmesi için demet merkezlerinin kullanılması*
- Uygulamalar
  - Örüntü tanıma
  - Görüntü işleme
  - Ekonomi - *Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs*
  - Aykırılıkları belirleme
  - WWW
    - Doküman demetleme
    - Kullanıcı davranışlarını demetleme
    - Kullanıcıları demetleme

Veri Madenciliğinde Demetlemenin Gereklilikleri

- Ölçeklenebilirlik
- Farklı tipteki ve niteliklerden oluşan nesneleri demetleme
- Farklı şekillerdeki demetleri oluşturabilme
- En az sayıda giriş parametresi gereksinimi
- Hatalı veriler ve aykırılıklardan en az etkilenme
- Çok boyutlu veriler üzerinde çalışma
- Sonucun yorumlanabilir ve anlaşılabilir olması

Kalite: İyi demetleme nedir?

- İyi bir demetleme yöntemiyle elde edilen demetlerin özellikleri
  - aynı demet içindeki nesneler arası benzerlik fazla
  - farklı demetlerde bulunan nesneler arası benzerlik az
- Oluşan demetlerin kalitesi seçilen benzerlik ölçütüne ve bu ölçütün gerçeklenmesine bağlı

Kalite: İyi demetleme nedir?

- Uzaklık / Benzerlik nesnelerin nitelik tipine göre değişir
  - Nesneler arası benzerlik: *s(i,j)*
  - Nesneler arası uzaklık: *d(i,j)* = 1 – *s(i,j)*
- İyi bir demetleme yöntemi veri içinde gizlenmiş örüntüleri bulabilmeli
- Veriyi gruplama için uygun demetleme kriteri bulunmalı
  - demetleme = aynı demetteki nesneler arası benzerliği en büyüten, farklı demetlerdeki nesneler arası benzerliği en küçülten fonksiyon

Veri Yapıları

- Veri matrisi
  n veri sayısı
  p nitelik sayısı
- Farklılık matrisi
  İki veri arasındaki
  Uzaklık (*d(i,j)*)

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix}$$

Veriler arası benzerlik ve farklılık ölçme

- En çok kullanılanlar arasında: **Minkowski** *uzaklığı:*

$$d(i,j) = \sqrt[q]{(|x_{i_1} - x_{j_1}|^q + |x_{i_2} - x_{j_2}|^q + ... + |x_{i_p} - x_{j_p}|^q)}$$

$i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ and $j = (x_{j1}, x_{j2}, \ldots, x_{jp})$ p-boyutlu iki veri. q pozitif bir tamsayı

- Eğer $q = 1$ ise *d'ye* **Manhattan** uzaklığı denir

$$d(i,j) = |x_{i_1} - x_{j_1}| + |x_{i_2} - x_{j_2}| + ... + |x_{i_p} - x_{j_p}|$$

Veriler arası benzerlik ve farklılık ölçme (devam)

- Eğer $q = 2$, *d'ye* **Öklid (*Euclidean*)** uzaklığı denir

$$d(i,j) = \sqrt{(|x_{i_1} - x_{j_1}|^2 + |x_{i_2} - x_{j_2}|^2 + ... + |x_{i_p} - x_{j_p}|^2)}$$

  - Öklid uzaklığı
    - $d(i,j) \geq 0$
    - $d(i,i) = 0$
    - $d(i,j) = d(j,i)$
    - $d(i,j) \leq d(i,k) + d(k,j)$

Demetler arası uzaklık ölçme

- Tek (Single) link: farklı demetlerdeki herhangi iki eleman arasındaki en küçük uzaklık, i.e., $dis(K_i, K_j) = min(t_{ip}, t_{jq})$
- Tam (Complete) link: farklı demetlerdeki herhangi iki eleman arasındaki en büyük uzaklık, i.e., $dis(K_i, K_j) = max(t_{ip}, t_{jq})$
- Ortalama (Average): farklı demetlerdeki elemanlar arasındaki ortalama uzaklık, i.e., $dis(K_i, K_j) = avg(t_{ip}, t_{jq})$
- **Centroid**: iki demetin centroid'lerinin arasındaki uzaklık,

  i.e., $dis(K_i, K_j) = dis(C_i, C_j)$
- **Medoid**: iki demetin medoid'lerinin arasındaki uzaklık,

  i.e., $dis(K_i, K_j) = dis(M_i, M_j)$
  - Medoid: one chosen, centrally located object in the cluster

Centroid, Radius and Diameter

- Centroid: kümenin merkezi $$C_m = \frac{\sum_{i=1}^{N}(t_{ip})}{N}$$
- Radius/yarıçap: square root of average squared distance from any point of the cluster to its centroid $$R_m = \sqrt{\frac{\sum_{i=1}^{N}(t_{ip} - c_m)^2}{N}}$$
- Diameter/çap: square root of average mean squared distance between all pairs of points in the cluster $$D_m = \sqrt{\frac{\sum_{i=1}^{N}\sum_{i=1}^{N}(t_{ip} - t_{iq})^2}{N(N-1)}}$$

Temel Demetleme Yaklaşımları

- **Bölünmeli yöntemler**: Veriyi bölerek, her grubu belirlenmiş bir kritere göre değerlendirir
- **Hiyerarşik yöntemler**: Veri kümelerini (ya da nesneleri) önceden belirlenmiş bir kritere göre hiyerarşik olarak ayırır
- **Yoğunluk tabanlı yöntemler**: Nesnelerin yoğunluğuna göre demetleri oluşturur
- **Model tabanlı yöntemler**: Her demetin bir modele uyduğu varsayılır. Amaç bu modellere uyan verileri gruplamak

Bölünmeli yöntemler

- **Amaç:** n nesneden oluşan bir veri kümesini (D) k (k≤n) demete ayırmak
  - her demette en az bir nesne bulunmalı
  - her nesne sadece bir demette bulunmalı
- **Yöntem:** Demetleme kriterini en çok büyütecek şekilde D veri kümesi k gruba ayırma
  - Global çözüm: Mümkün olan tüm gruplamaları yaparak en iyisini seçmek (NP karmaşık)
  - Sezgisel çözüm: **k-means** ve **k-medoids**
    - k-means (MacQueen'67): Her demet kendi merkezi ile temsil edilir
    - k-medoids veya PAM (Partition around medoids) (Kaufman & Rousseeuw'87): Her demet, demette bulunan bir nesne ile temsil edilir

K-means Demetleme

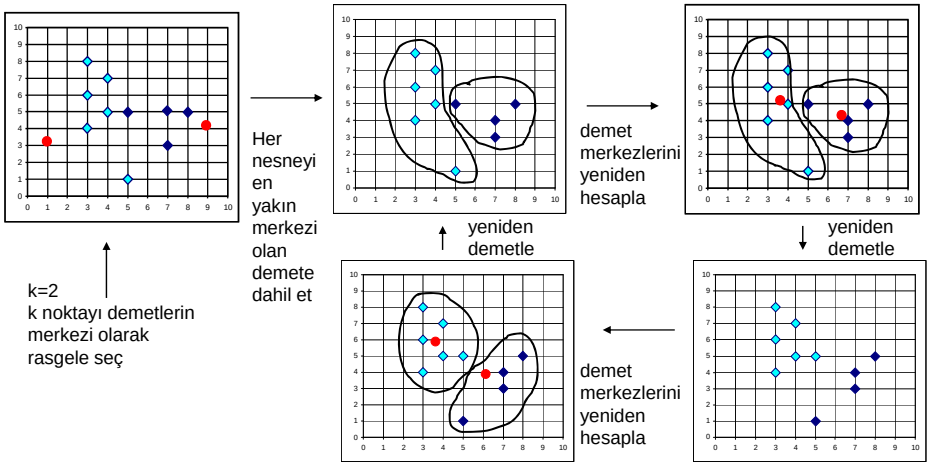
- Bilinen bir ***k*** değeri için ***k-means*** demetleme algoritmasının 4 aşaması vardır:
  - Veri kümesi k altkümeye ayrılır (her demet bir altküme)
  - Her demetin ortalaması hesaplanır: merkez nokta (demetteki nesnelerin niteliklerinin ortalaması)
  - Her nesne en yakın merkez noktanın olduğu demete dahil edilir
  - Nesnelerin demetlenmesinde değişiklik olmayana kadar adım 2'ye geri dönülür.

Algorithm K-Means

1: K tane rasgele noktayı demet merkezi (centroid) olarak seç.
2: **Repeat:**
3: Her bir veriyi kendine en yakın centroide atayarak K demet oluştur.
4: Oluşan demetlerin centroidlerini hesapla.
5: **Until:** Centroidler değişmeyene kadar.

K-means Demetleme

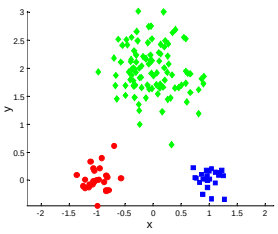
- Örnek:



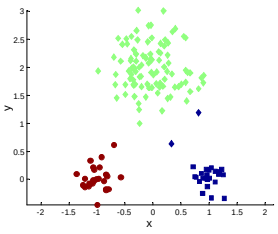
K-means Demetleme

- Demet sayısının belirlenmesi gerekir
- Başlangıçta demet merkezleri rasgele belirlenir bu nedenle her uygulamada farklı demetler oluşabilir
- Benzerlik Öklid uzaklığı 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

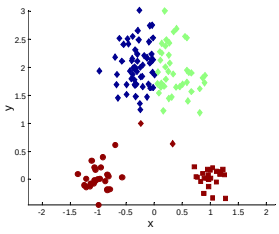
Başlangıç centroid seçiminin önemi



Orijinal Veri

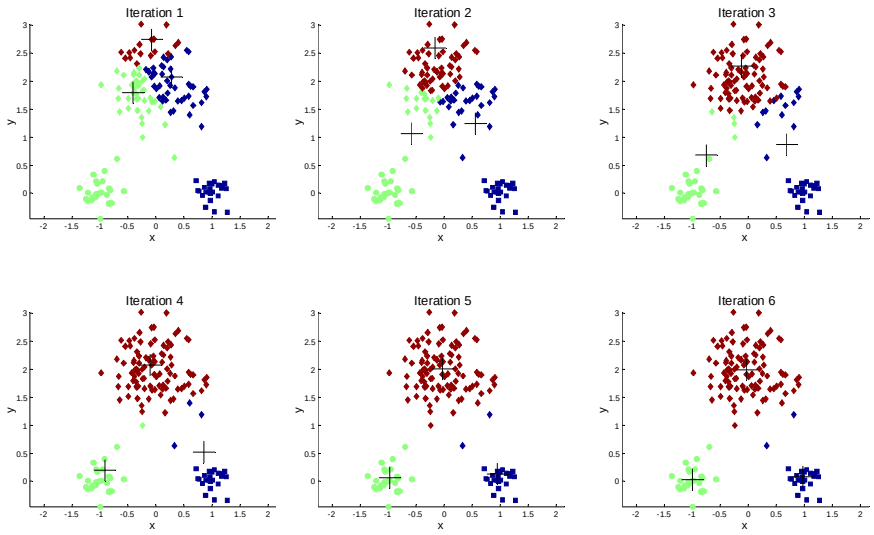


Optimal Demetleme

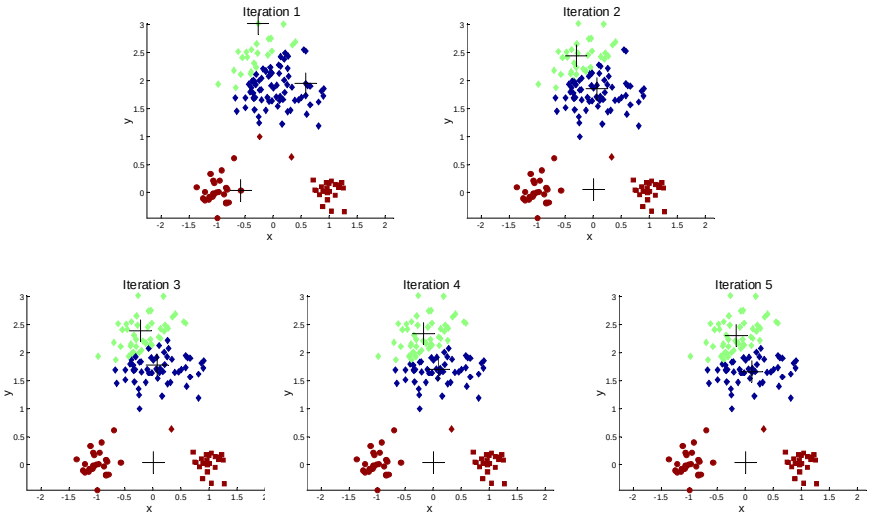


Sub-optimal Demetleme

Başlangıç centroid seçiminin önemi



Başlangıç centroid seçiminin önemi



K-means Demetleme Değerlendirme

- Birden çok sonuç oluşur.
- Yaygın olarak kullanılan yöntem hataların karelerinin toplamı (Sum of Squared Error SSE)

$$\sum_{m=1}^{k} \sum_{t_{mi} \in Km} (C_m - t_{mi})^2$$

- Nesnelerin bulundukları demetin merkez noktalarına olan uzaklıklarının karelerinin toplamı
- Hataların karelerinin toplamını azaltmak için k demet sayısı artırılabilir
- Başlangıç için farklı merkez noktaları seçerek farklı demetlemeler oluşturulur
- En az SSE değerini sahip olan demetleme seçilir

Avantaj - Dezavantaj

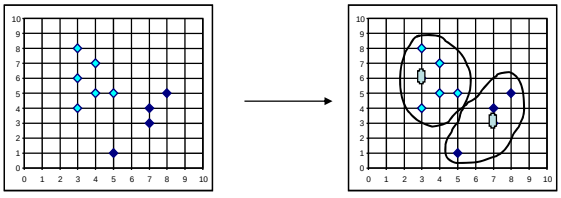
- Gerçeklemesi kolay
- Karmaşıklığı diğer demetleme yöntemlerine göre az
- K-Means algoritması bazı durumlarda iyi sonuç vermeyebilir
  - Veri grupları farklı boyutlarda ise
  - Veri gruplarının yoğunlukları farklı ise
  - Veri gruplarının şekli küresel değilse
  - **Veri içinde aykırılıklar (outliers)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

Outlier sorunu

- k-means algoritması sapan verilerden (outlier) etkileniyor!
  - Çok büyük ya da çok küçük bir veri dağılımını etkileyebilir
- Çözüm
  - **K-Medoids:** Veri ortalamasını kullanmak yerine medoid kullanılabilir.
  - Medoid demet içerisinde en merkezi pozisyondaki veridir.

Örnek

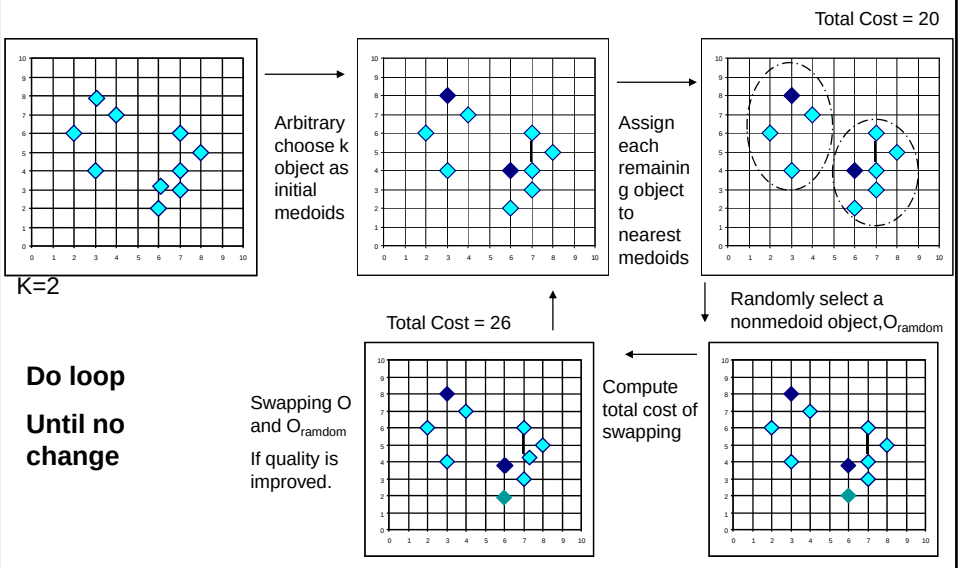
- Her demeti temsil etmek için demet içinde orta nokta olan nesne seçilir.
  - 1, 3, 5, 7, 9 ortalama: 5
  - 1, 3, 5, 7, 1009 ortalama: 205
  - 1, 3, 5, 7, 1009 orta nokta: 5



K-Medoids Demetleme

- PAM (Partitioning Around Medoids, 1987)
  1. Başlangıçta *k* adet nesne demetleri temsil etmek üzere rasgele seçilir x_{ik}
  2. Kalan nesneler en yakın merkez nesnenin bulunduğu demete dahil edilir
  3. Merkez nesne olmayan rasgele bir nesne seçilir x_{rk}
  4. x_{rk} merkez nesne olursa toplam karesel hatanın ne kadar değiştiğini bulunur
  5. Eğer değişim negatifse yani hata da azalmaya sebep oluyorsa, x_{rk} merkez nesne olarak atanır.
  6. Demetlerde değişiklik oluşmayana kadar 3. adıma geri gidilir.

K-Medoids Algorithm (PAM)



PAM'daki sorunlar

- Pam k-means ile karşılaştırıldığında daha güvenilir bir algoritma
  - Medoid sapan verilerden (outliers) ortalamaya göre daha az etkilenir
- Küçük veri kümeleri için iyi sonuç verebilir, ancak büyük veri kümeleri için uygun değil
  - Her iterasyon için karmaşıklık : $O(k(n-k)^2)$

- CLARA(Clustering LARge Applications)
- CLARANS (Ng & Han, 1994)

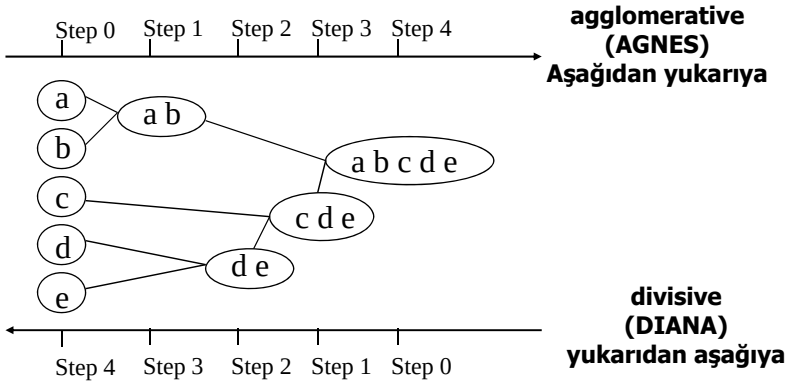
Temel Demetleme Yaklaşımları

- **Bölünmeli yöntemler**: Veriyi bölerek, her grubu belirlenmiş bir kritere göre değerlendirir
- **Hiyerarşik yöntemler**: Veri kümelerini (ya da nesneleri) önceden belirlenmiş bir kritere göre hiyerarşik olarak ayırır
- **Yoğunluk tabanlı yöntemler**: Nesnelerin yoğunluğuna göre demetleri oluşturur
- **Model tabanlı yöntemler**: Her demetin bir modele uyduğu varsayılır. Amaç bu modellere uyan verileri gruplamak

Hiyerarşik Demetleme

- İki genel metod
  - Agglomerative:
    - Her veriyi ayrı olarak düşün
    - Her aşamada yakın verileri birleştir
  - Divisive:
    - Tüm verileri bir olarak düşün
    - Her aşamada uzak olanları ayır
- Benzerlik ya da uzaklık matrisi kullanılır
  - Merge or split one cluster at a time
- Demet sayısının belirlenmesine gerek yok
  - Sonlanma kriteri belirlenmesi gerekiyor

Hiyerarşik Demetleme

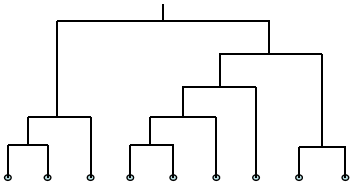


agglomerative (AGNES) Aşağıdan yukarıya

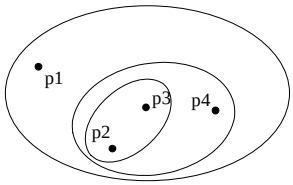
divisive (DIANA) yukarıdan aşağıya

Hiyerarşik Demetleme

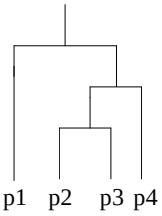
- **Dendogram:** Demetler hiyerarşik olarak ağaç yapısı şeklinde görüntülenebilir
- Ara düğümler çocuk düğümlerdeki demetlerin birleşmesiyle elde edilir
  - Kök: bütün nesnelerden oluşan tek demet
  - Yapraklar: bir nesneden oluşan demetler
- Dendogram istenen seviyede kesilerek demetler elde edilir



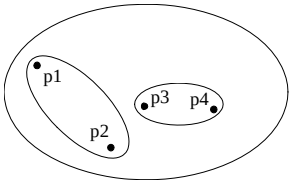
Hiyerarşik Demetleme



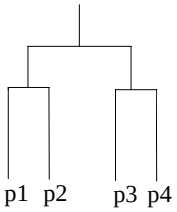
Traditional Hierarchical Clustering



Traditional Dendrogram



Non-traditional Hierarchical Clustering



Non-traditional Dendrogram

Agglomerative Demetleme Algoritması

- En çok kullanılan hiyeraşik demetleme algoritması
  - Oldukça basit

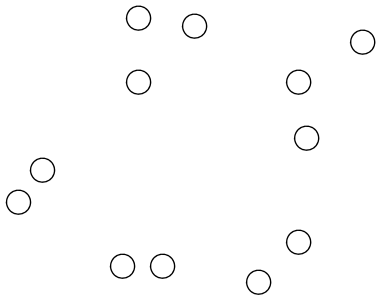
Algorithm Agglomerative Demetleme

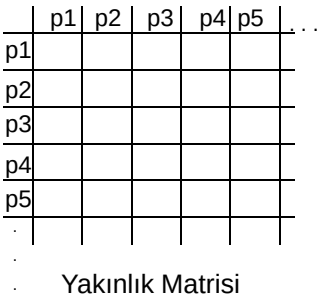
1: Yakınlık (proximity) matrisini hesapla.
2: Her veri noktası bir demet olsun.
3: **Repeat:**
4: En yakın iki demeti birleştir.
5: Yakınlık (proximity) matrisini güncelle.
6: **Until:** Sadece tek bir demet oluşana kadar.

- En önemli işlem iki demet arasındaki yakınlığın bulunması

İlk aşama

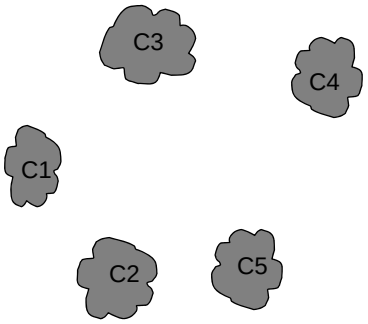
- Her veri bir demet





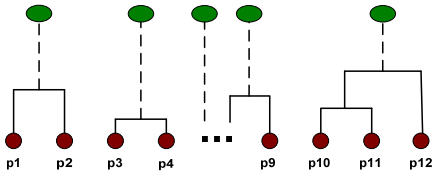
Ara işlemler

- Bir kaç işlemden sonra demetler oluşur.



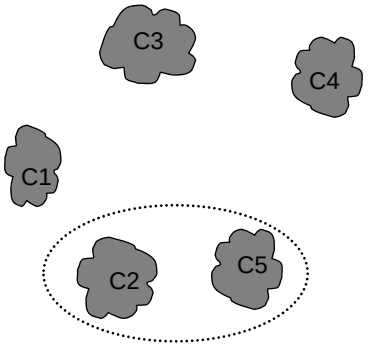
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Yakınlık Matrisi



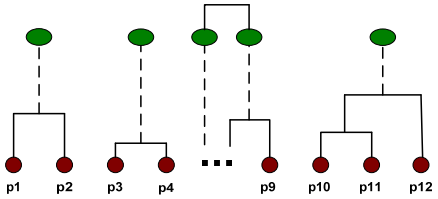
Ara işlemler

- En yakın demetler C2 ve C5 birleştirlecek



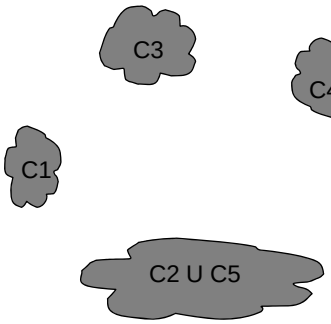
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Yakınlık Matrisi



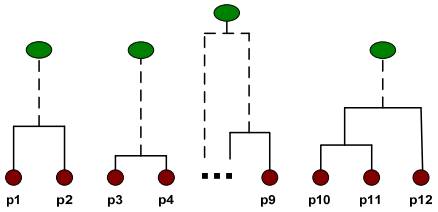
Birleşmeden sonra

- Yakınlık matrisi nasıl güncellenecek

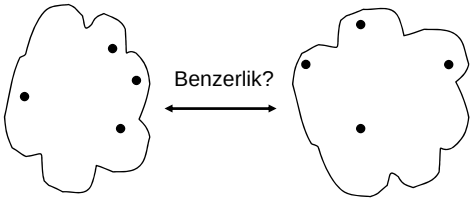


	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Yakınlık matrisi



Demetler arası benzerlik (yakınlık)

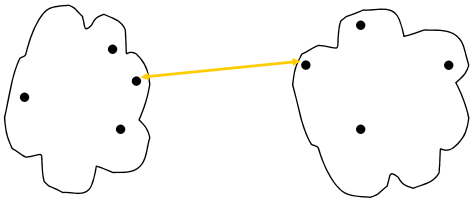


- MIN
- MAX
- Grup Ortalaması
- Centroidler arası uzaklık

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

Yakınlık Matrisi

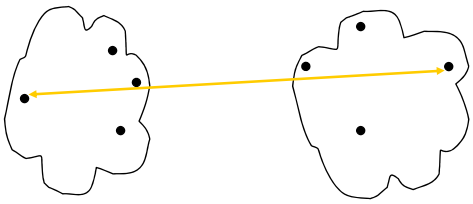
Demetler arası benzerlik (yakınlık)



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

- MIN
- MAX
- Grup Ortalaması
- Centroidler arası uzaklık

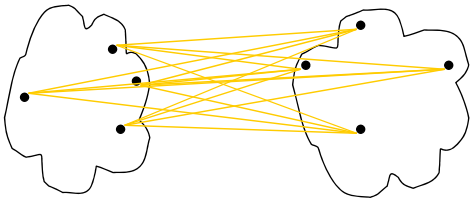
Demetler arası benzerlik (yakınlık)



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

- MIN
- MAX
- Grup Ortalaması
- Centroidler arası uzaklık

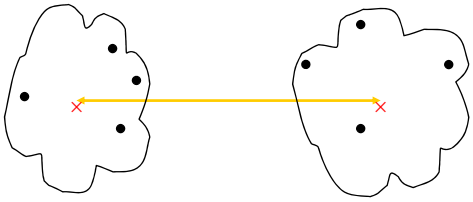
Demetler arası benzerlik (yakınlık)



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

- MIN
- MAX
- Grup Ortalaması
- Centroidler arası uzaklık

Demetler arası benzerlik (yakınlık)



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

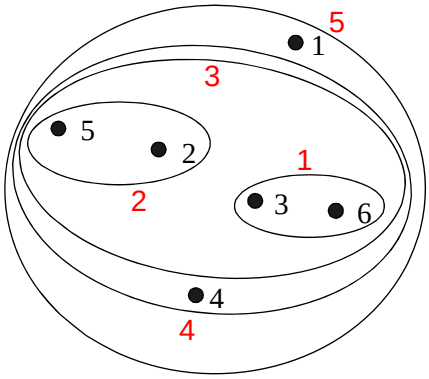
- MIN
- MAX
- Grup Ortalaması
- Centroidler arası uzaklık

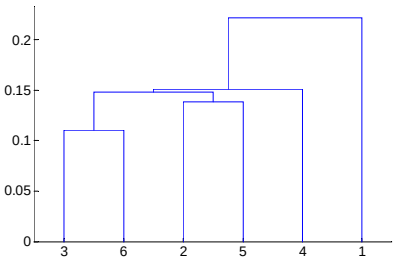
Örnek

	p1	p2	p3	p4	p5	p6
p1	0	0,24	0,22	0,37	0,34	0,23
p2	0,24	0	0,15	0,2	0,14	0,25
p3	0,22	0,15	0	0,15	0,28	0,11
p4	0,37	0,2	0,15	0	0,29	0,22
p5	0,34	0,14	0,28	0,29	0	0,39
p6	0,23	0,25	0,11	0,22	0,39	0

Uzaklık matrisi (Öklid)

Hierarchical Clustering: MIN



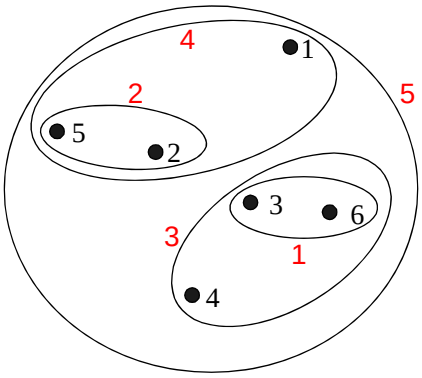


Nested Clusters

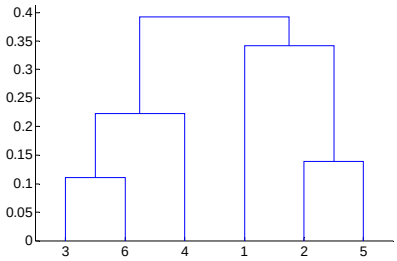
Dendrogram

dist({3,6},{2,5})=min(dist(3,2);dist(6,2);dist(3,5);dist(6,5))=min(0,15;0,25;0,28;0,39)=0,15

Hierarchical Clustering: MAX



Nested Clusters



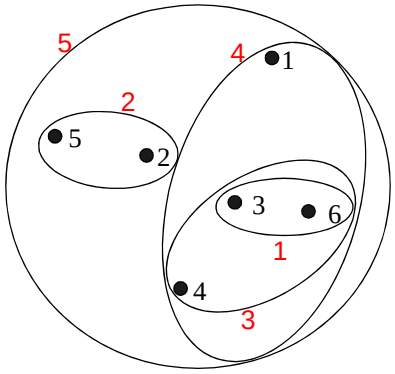
Dendrogram

dist({3,6},{4})=max(dist(3,4),dist(6,4))=max(0,15 0,22)=0,22

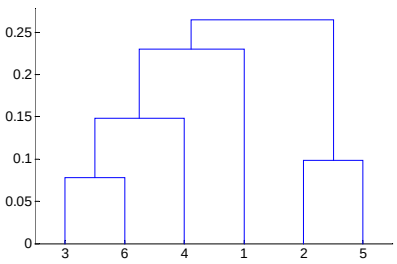
dist({3,6},{2,5})=max(dist(3,2),dist(6,2),dist(3,5),dist(6,5))=max(0,15 0,25 0,28 0,39)=0,39

dist({3,6},{1})=max(dist(3,1),dist(6,1))=max(0,22 0,23)=0,23

Hierarchical Clustering: Group Average



Nested Clusters



Dendrogram

Hierarchical Clustering: Group Average

- Compromise between Single (min) and Complete (max) Link
- Strengths
  - Less susceptible to noise and outliers
- Limitations
  - Biased towards globular clusters