Hafta 04 - Karar Ağaçları/Kümeleme

SİB 552 - Siber Güvenlik İçin Veri Madenciliği

Bilgisayar Mühendisliği
Siber Güvenlik Yüksek Lisans Programı

Dr. Ferhat Özgür Çatak
ozgur.catak@tubitak.gov.tr

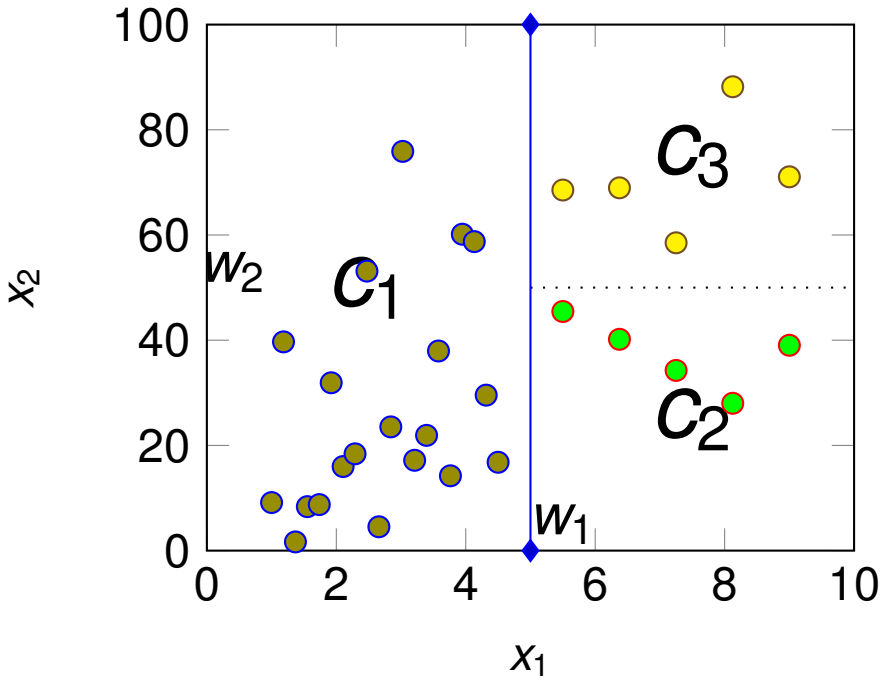
Gebze Teknik Üniversitesi
2019 - Bahar

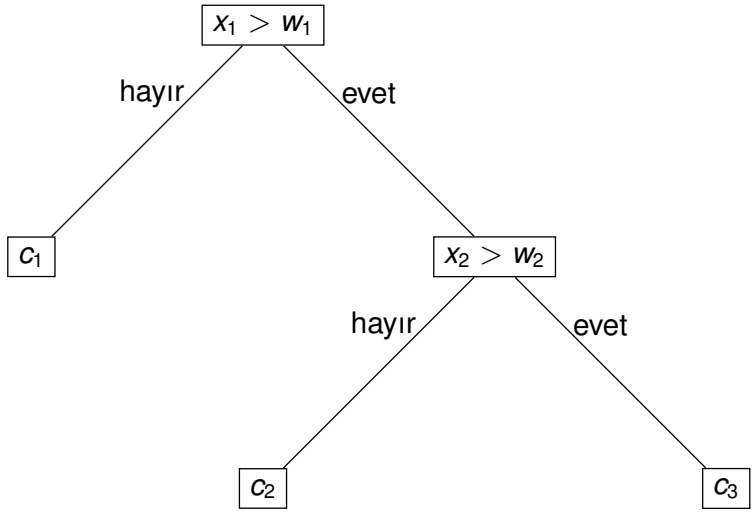
İçindekiler

İçindekiler

Giriş

- Böl-ve-fethet stratejisine dayalı hiyerarşik veri yapılarıdır.
- Girdi uzayını lokal bölgelere yinelemeli olarak ayıran hiyerarşik model.
- Bir karar ağacı modeli, *karar düğümü* ve *hedef yapraklarından* oluşmaktadır.
- Her bir *karar düğümü* $f_m(\mathbf{x})$ şeklinde bir test fonksiyonuna sahiptir.





Tek Değişkenli Ağaçlar
Univariate Trees

Tek Değişkenli Ağaçlar

- Her bir düğümde sadece bir öznitelik kontrol edilir.
- Düğümde kullanılan öznitelik, $\mathbf{x}_i$, kesikli olması durumunda n adet farklı ayrıştırma gerçekleştirilmelidir.
  - Örnek: öznitelik servis $\Rightarrow$ {http, ftp, ssh} olması durumunda 3 farklı dallanma olmalıdır.
- Düğümde kullanılan öznitelik, $\mathbf{x}_i$, sürekli olması durumunda ise kesikli hale getirilmelidir.
  - Örnek: paket sayısı $\Rightarrow f_m(\mathbf{x}_i) : \mathbf{x}_i > w_m$
  - w_m değeri, uygun bir yöntemle seçilmiş olan eşik değeridir.
  - Karar düğümü, girdi uzayını ikiye bölmektedir: $L_m = \{\mathbf{x} | x_j > w_m\}$ ve $R_m = \{\mathbf{x} | x_j \leq w_m\}$
  - İkili ayrıştırma: *binary split*

Sınıflandırma Ağaçları I
Classification Trees

Sınıflandırma Ağaçları

- Bir ayrıştırmanın ne kadar doğru yapıldığının ölçümü: Belirsizlik ölçümü (Impurity Measure)
- Eğer bir ayrıştırma yapıldıktan sonra, bir dal içinde yer alan örneklerin tamamı aynı sınıfa etiketine sahip olması durumunda, bu ayrıştırma *saftır (pure)*.
- Düğüm m, bu düğümde yer alan örnek (satır) sayısı N_m, ve sınıf C_i'ye ait olan örneklerin sayısı N_m^i olsun. Bu durumda bir örneğin, $\mathbf{x}$, m düğümüne eriştiğinde bir C_i sınıfına ait olma olasılığı:

$$P(C_i|\mathbf{x}, m) = \frac{N_m^i}{N_m} \tag{1}$$

  - Elinizde 5 adet zararlı yazılım ($\mathbf{y}_m = \{ransomware, trojan, trojan, ransomware, backdoor\}$) olduğunu kabul edin. Bu durumda $N_m = 5$. Seçtiğiniz bir yazılımın, $\mathbf{x}$, ransomware ($C_i = ransomware$) olma olasılığı nedir? $\mathbf{x}$ ve m bilindiğine göre C_i'nin olasılığı - koşullu olasılık.

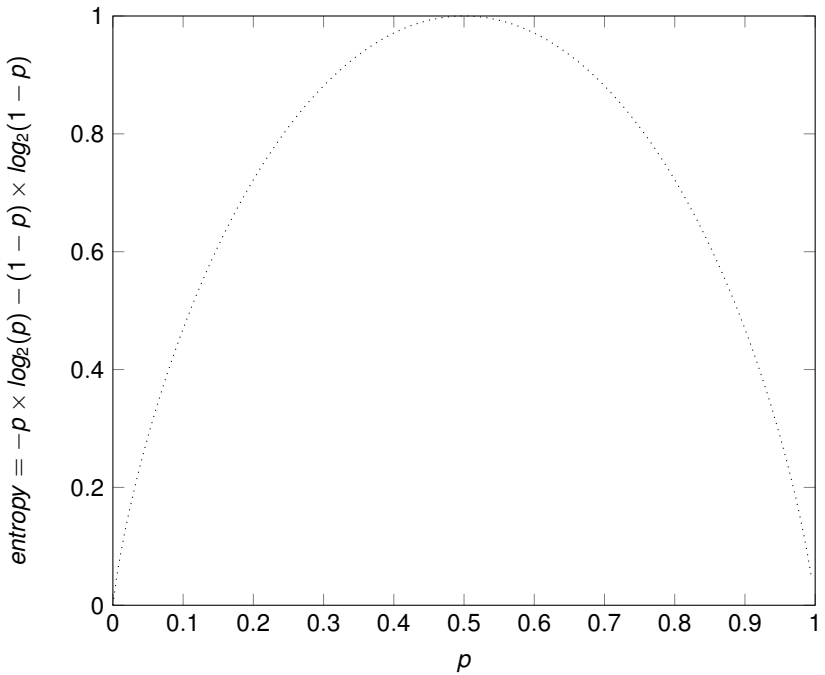
$$P(C_i|\mathbf{x}, m) = \frac{N_m^i}{N_m} = \frac{2}{5} = 0.4$$

Sınıflandırma Ağaçları II
Classification Trees

Entropy

- Düzensizlik ölçümlerinden bir tanesidir.

$$H_m = -\sum_{i=1}^{K} p_i log_2(p_i) \tag{2}$$



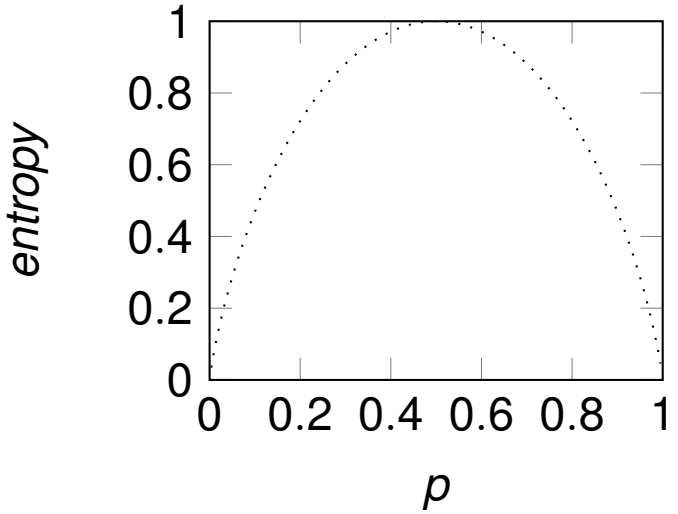
Sınıflandırma Ağaçları III
Classification Trees

Düzensizlik ölçüm yöntemleri

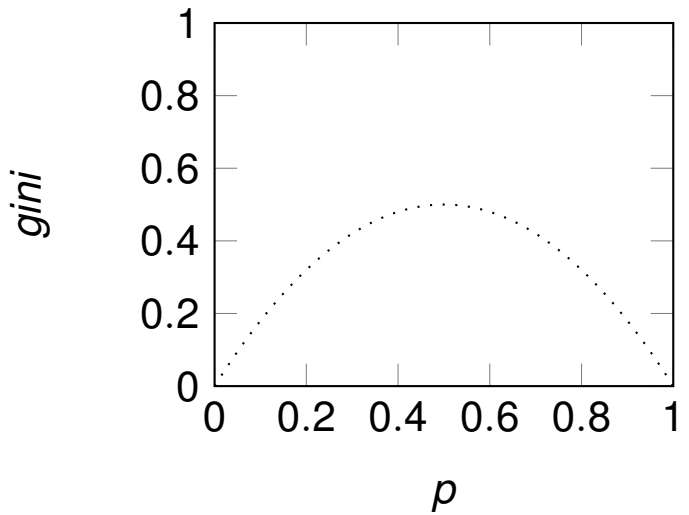
- Entropy tek yöntem değildir.
- İki sınıflı problem için 1. sınıf için olasılık, $p_1 = p$, 2. sınıf için olasılık $p_2 = 1 - p_1 = 1 - p$
- Genel çözüm: $\phi(p, 1-p)$ negative olmayan fonksiyon tanımlanmalıdır.
  - Negative olmayan fonksiyon: $\phi \geq 0$, $\phi \in \mathbb{R}_+$
- Özellikleri:
  - $\phi(0,1) = \phi(1,0) = 0$
  - $\phi(p, 1-p)$ fonksiyonu $[0, 1/2]$ aralığında artış, $[1/2, 1]$ aralığında azalış göstermelidir.
- Yöntemler:
  - **Entropy**: $\phi(p, 1-p) = -p \times log_2(p) - (1-p) \times log_2(1-p)$
  - **Gini**: $\phi(p, 1-p) = 2 \times p \times (1-p)$
  - **Misclassification Error**: $\phi(p, 1-p) = 1 - max(p, 1-p)$

Sınıflandırma Ağaçları IV
Classification Trees

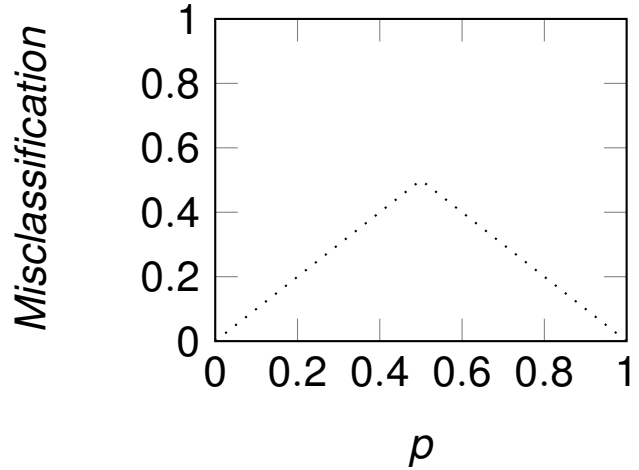
$$\phi(p, 1-p) = -plog_2p-(1-p)log_2(1-p)$$



$$\phi(p, 1-p) = 2 \times p \times (1-p)$$



$$\phi(p, 1-p) = 1 - max(p, 1-p)$$



Sınıflandırma Ağaçları V
Classification Trees

Düzensizlik (Impurity)

- Bir düğüm m saf (pure) değilse, düzensizliği en fazla azaltacak şekilde sahip olduğu örnekler ayrıştırışmalıdır.
  - Bir düğümün sahip olduğu örneklerin sınıf etiketleri aynı ise bu düğüm saftır.
- Düğüm m ait olan örneklerin içinde j dalının örnek sayısı N_{mj} olsun. Bu durumda j dalının içinde C_i sınıfının olasılığı

$$P(C_i|\mathbf{x}, m, j) = p^i_{mj} = \frac{N^i_{mj}}{N_{mj}}$$

- Ayrıştırma olduktan sonra toplam düzensizlik

$$\mathcal{I}_m = -\sum_{j=1}^{N} \frac{N_{mj}}{N_m} \sum_{i=1}^{K} p^i_{mj} log_2(p^i_{mj}) \quad (3)$$

Karar Ağacı Algoritmaları I

ID3 (Iterative Dichotomiser 3)

- 1986 yılında Ross Quinlan tarafından önerilmiştir.
- Kategorik hedefler (sınıflar) için kategorik öznitelikleri kullanmaktadır.
- Bilgi kazancının en fazla olacağı nitelik seçilerek aşağıya doğru devam etmektedir.

Karar Ağacı Algoritmaları II

C4.5

- 1993 yılında Ross Quinlan tarafından önerilmiştir.
- Sayısal değerler içerebilir. Bir eşik değeri seçilerek aralığın orta noktası bulunur.
- Sayısal değere sahip olan nitelik sıralanır $\mathbf{x}_f = \{v_1, v_2, \cdots, v_n\}$. Eşik değeri ortanca değer seçilebilir
- **Örnek**:
  - bir niteliğin içerdiği değerler sıralansın. $\mathbf{x}_f = \{15, 20, 25, 30, 35, 40\}$.
  - $\mathbf{x}_f \in \mathbb{R}^6$ olması nedeniyle 25 ve 30 ortanca değerlerdir. Eşik noktası $t_f = \frac{25+30}{2} = 27.5$.
  - Bu nitelik değerleri $\mathbf{x}_f = \{\leq 27.5, \leq 27.5, \leq 27.5, > 27.5, > 27.5, > 27.5\}$

Karar Ağacı Algoritmaları III

CART (Classification And Regression Trees)

- C4.5 algoritmasına oldukça benzer.
- Sayısal değerlerin eşik değerleri için bilgi kazanım değerinin en yüksek olduğu değer seçilir.
- **scikit-learn** kütüphanesi CART kullanmaktadır.
- Sayısal değerlerin ayrıştırılabilmesi için en yüksek kazanç noktasını bulmaktadır.

Python

Python

```
class sklearn.tree.DecisionTreeClassifier(
criterion='gini', splitter='best', max_depth=None,
min_samples_split=2, min_samples_leaf=1,
min_weight_fraction_leaf=0.0, max_features=None,
random_state=None, max_leaf_nodes=None,
min_impurity_decrease=0.0, min_impurity_split=None,
class_weight=None, presort=False)
```

Lab

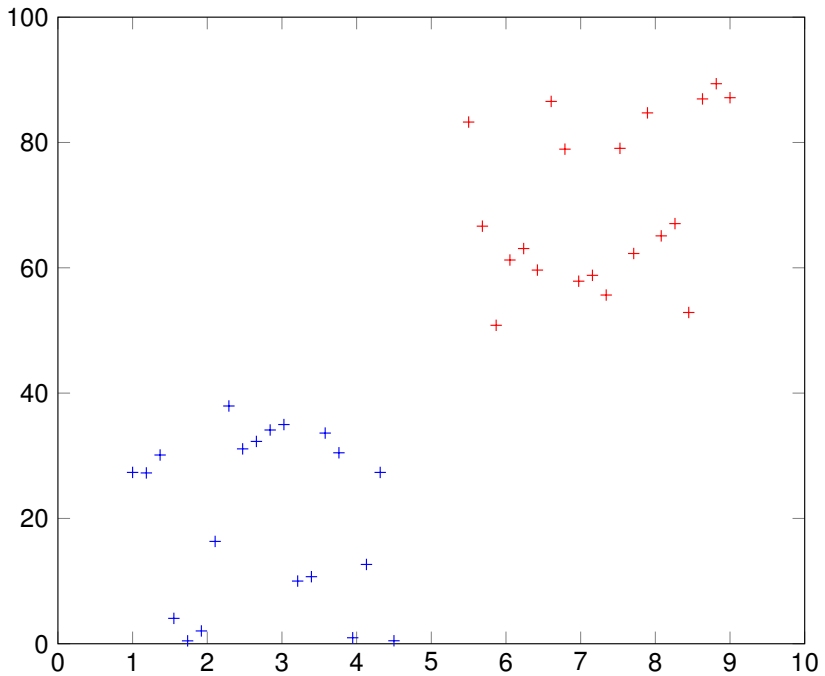
Lab - 1

İçindekiler

Giriş

Kümeleme

- Bir veri kümesi içerisinde yer alan benzer nesnelerin aynı gruplarda yer alacak şekilde ayrıştırılmasıdır.
- Birçok gerçek dünya veri kümesinde örneklem tek bir öbek yerine bir kaç farklı öbekten gelebilmektedir.



Kümelemenin Amacı

Amaç

- Bir veri kümesi içerisinde yer alan benzer nesnelerin aynı gruplarda yer alacak şekilde ayrıştırılmasıdır.
- **Örnekler**:
  - Bir hastalığa neden olan genlerin gruplanması
  - Benzer ilgileri olan kişilerin gruplanması

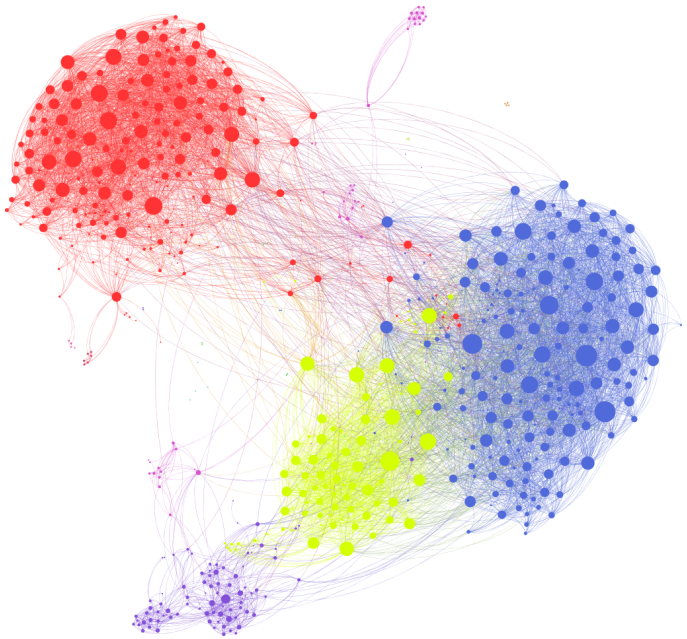
Table: User-Item Matrix

	$Film_1$	$Film_1$	$\cdots$	$Film_n$
$user_1$	10	1	$\cdots$	8
$user_2$	3	9	$\cdots$	2
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$user_n$	10	1	$\cdots$	8

Kümelemenin Uygulama Alanları

Uygulama Alanları

- Veri kümesinde genel karakteristiğini anlamak
- Veriyi görselleştirmek
- **Örnekler**
  - Sosyal ağlarda toplulukların bulunması



Centroids ve Medoids

Centroid

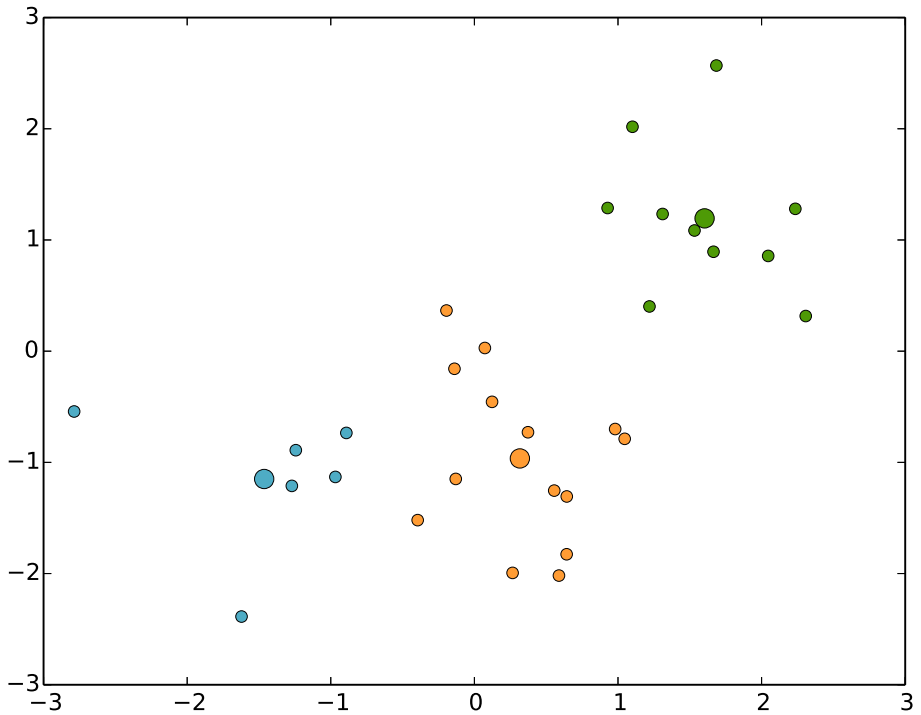
Kümedeki noktaların ortalaması.

$$\mu = \frac{1}{|C|} \sum_{\mathbf{x} \in C} \mathbf{x} \tag{4}$$

Medoid

Centroide en yakın nokta

$$m = \underset{\mathbf{x} \in C}{arg\,mind}(\mathbf{x}, \mu) \tag{5}$$



K-Ortalamalar Kümeleme
K-means clustering

K-Ortalamalar

- Her bir küme içinde varyansı düşürmek

$$Var(C) = \frac{1}{|C|} \sum_{\mathbf{x} \in C} ||\mathbf{x} - \mu_c||^2$$

- Parametre: kümesi sayısı k
- Yüksek sayıda örnek içeren veri kümeleri için uygun bir yöntemdir.
- Algoritma 4 adımdan oluşmaktadır.
  - Rassal k adet centroid belirle
  - Veri kümesinde yer alan her bir örneğe en yakın centroide göre küme etiketi ata
  - Kümeler için de ortalama yöntemi ile yeni centroidleri hesapla
  - centroidlerde belirli bir eşik değerininden daha az değişim oluncaya kadar yukarıda yer alan adımları tekrarla.

Lab

Lab - 2

K-Means Kümeleme İçin Silhouette Analizi I

Silhouette Analizi

- Oluşturulan kümelerde yer alan verilerin tutarlığını ölçmek için kullanılan bir yöntemdir.
- $a(i)$: i noktası ile aynı kümede yer alan noktalar arasında bulunan uzaklığın ortalaması
  - $a(i)$ ne kadar küçük ise doğru bir küme ataması yapıldığı anlamındadır.
- $b(i)$: i'nin üye olmadığı diğer kümelerde yer alan noktalara olan uzaklığın en az değeri. En yakın komşuya olan ortalama uzaklık.

Silhouette:

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

$$s(i) = \begin{cases} 1 - a(i)/b(i) & if\ a(i) < b(i) \\ 0 & if\ a(i) = b(i) \\ b(i)/a(i) - 1 & if\ a(i) > b(i) \end{cases} \tag{6}$$

$$-1 \leq s(i) \leq 1$$

K-Means Kümeleme İçin Silhouette Analizi II

Silhouette

- $s(i)$ ifadesi $a(i) \ll b(i)$ için 1'e yaklaşır. $a(i)$ ifadesi küme içinde uzaklık olması nedeniyle, değerin düşük olması doğru bir kümede olduğu anlamına gelir.
- $b(i)$ büyük olması en yakın kümeye uzak olduğunu dolayısıyla doğru kümede olduğu anlamına gelir.
- Ortalama $s(i)$ ise bir kümede yer alan bütün örnekleri $s(i)$ değerlerinin ortalamasıdır.

Lab

Lab - 3

Hierarchical clustering I

Hierarchical clustering [1]

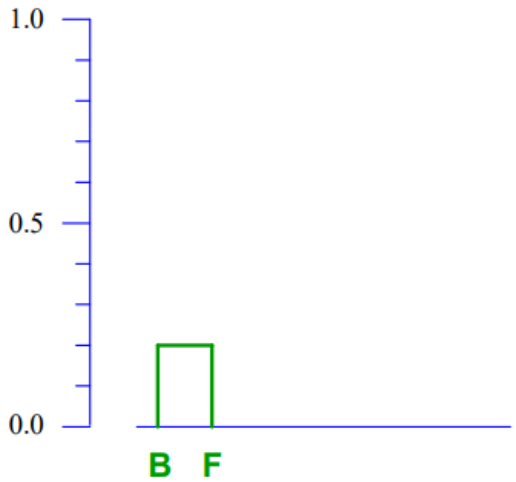
- Kümeleri bir araya getirerek ya da birbiri ile ayırarak oluşturan genel bir kümeleme algoritması ailesidir.
- Bu kümeler hiyerarşisi bir ağaç (veya dendrogram) olarak temsil edilir.
- Ağacın kökü (root), tüm örnekleri toplayan tekil kümedir (unique cluster), yapraklar sadece bir örnekle kümedir.

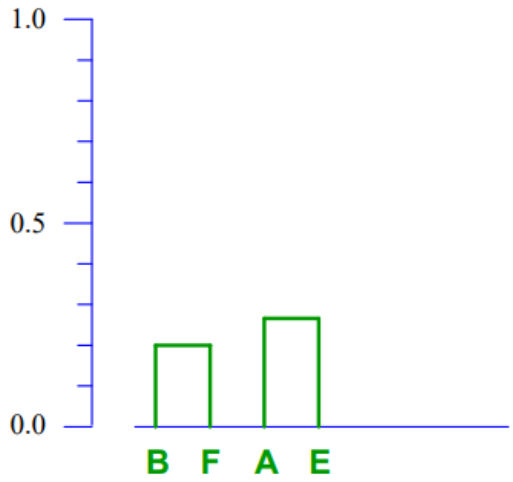
Hierarchical clustering II

samples	A	B	C	D	E	F	G
A	0	0.5000	0.4286	1.0000	0.2500	0.6250	0.3750
B	0.5000	0	0.7143	0.8333	0.6667	0.2000	0.7778
C	0.4286	0.7143	0	1.0000	0.4286	0.6667	0.3333
D	1.0000	0.8333	1.0000	0	1.0000	0.8000	0.8571
E	0.2500	0.6667	0.4286	1.0000	0	0.7778	0.3750
F	0.6250	0.2000	0.6667	0.8000	0.7778	0	0.7500
G	0.3750	0.7778	0.3333	0.8571	0.3750	0.7500	0

samples	A	(B,F)	C	D	E	G
A	0	0.6250	0.4286	1.0000	0.2500	0.3750
(B,F)	0.6250	0	0.7143	0.8333	0.7778	0.7778
C	0.4286	0.7143	0	1.0000	0.4286	0.3333
D	1.0000	0.8333	1.0000	0	1.0000	0.8571
E	0.2500	0.7778	0.4286	1.0000	0	0.3750
G	0.3750	0.7778	0.3333	0.8571	0.3750	0

Hierarchical clustering III

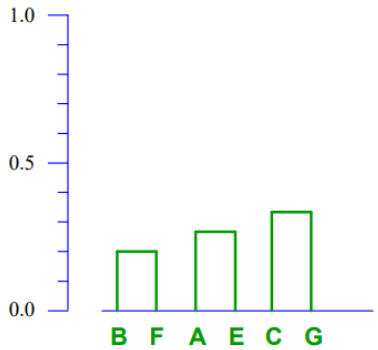


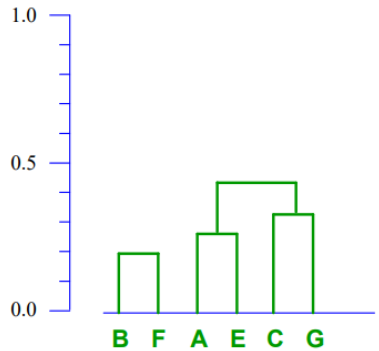


samples	(A,E)	(B,F)	C	D	G
(A,E)	0	0.7778	0.4286	1.0000	0.3750
(B,F)	0.7778	0	0.7143	0.8333	0.7778
C	0.4286	0.7143	0	1.0000	0.3333
D	1.0000	0.8333	1.0000	0	0.8571
G	0.3750	0.7778	0.3333	0.8571	0

Hierarchical clustering IV

samples	(A,E)	(B,F)	(C,G)	D
(A,E)	0	0.7778	0.4286	1.0000
(B,F)	0.7778	0	0.7778	0.8333
(C,G)	0.4286	0.7778	0	1.0000
D	1.0000	0.8333	1.0000	0

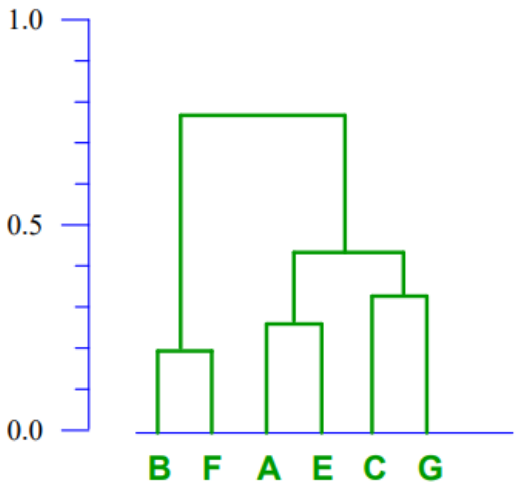


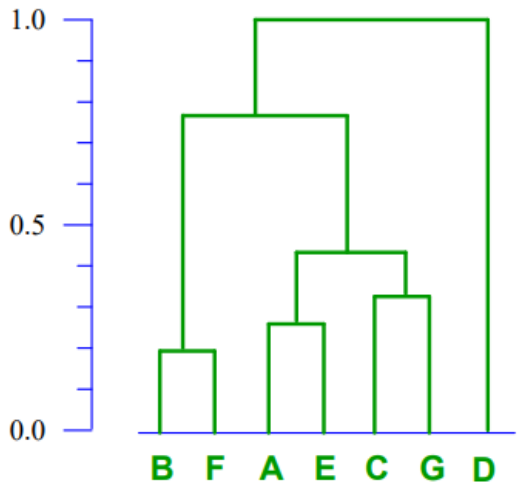


Hierarchical clustering V

samples	(A,E,C,G)	(B,F)	D
(A,E,C,G)	0	0.7778	1.0000
(B,F)	0.7778	0	0.8333
D	1.0000	0.8333	0

samples	(A,E,C,G,B,F)	D
(A,E,C,G,B,F)	0	1.0000
D	1.0000	0



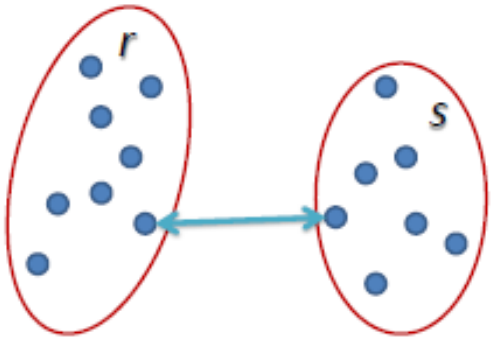


Hierarchical clustering VI

Single Linkage

- İki küme arasındaki uzaklık, kümeler içinde yer alan örnekler arasında bulunan en kısa uzaklıktır.

$$d(r,s) = \min_{\mathbf{x}_r \in r, \mathbf{x}_s \in s}(d(\mathbf{x}_r, \mathbf{x}_s))$$

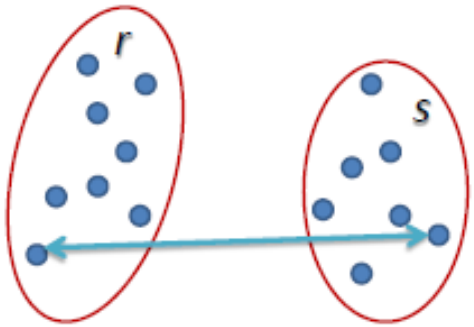


Hierarchical clustering VII

Complete Linkage

- İki küme arasındaki uzaklık, kümeler içinde yer alan örnekler arasında bulunan en uzun uzaklıktır.

$$d(r,s) = \underset{\mathbf{x}_r \in r, \mathbf{x}_s \in s}{max}(d(\mathbf{x}_r, \mathbf{x}_s))$$

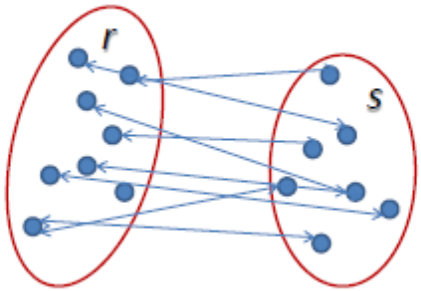


Hierarchical clustering VIII

Average Linkage

- İki küme arasındaki uzaklık, kümeler içinde yer alan örnekler arasında bulunan en uzun uzaklıktır.

$$d(r,s) = \frac{1}{|r||s|} \sum_{\mathbf{x}_{ri} \in r} \sum_{\mathbf{x}_{si} \in s} d(\mathbf{x}_{ri}, \mathbf{x}_{si})$$



[1] http://www.econ.upf.edu/ michael/stanford/maeb7.pdf

Lab

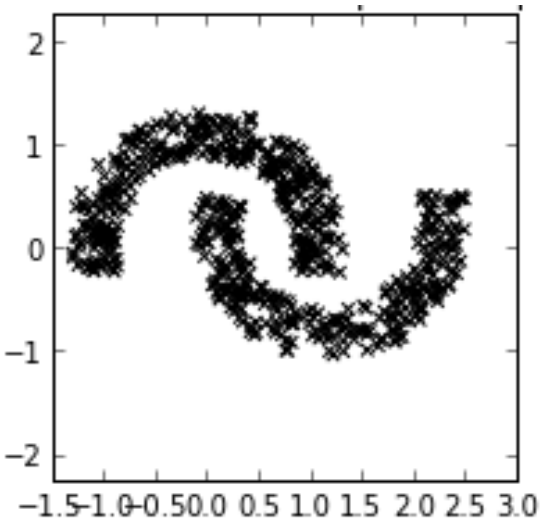
Lab - 4

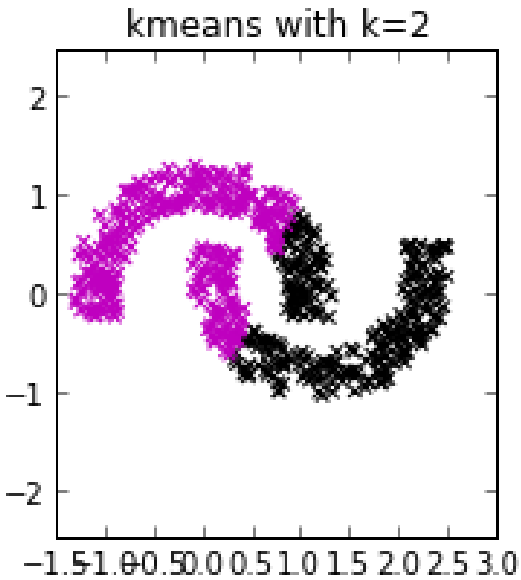
DBSCAN I

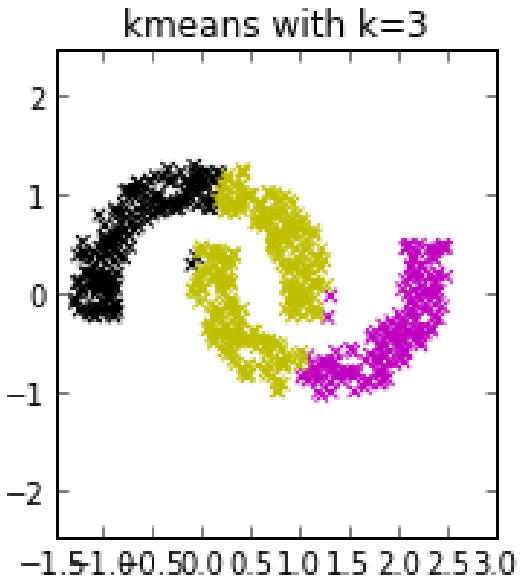
Density-based spatial clustering of applications with noise

Convex Set

- Euclid uzayında yer alan bir nesne convex olabilmesi için bütün noktalarının doğrusal bir çizgi ile birleşebilmelidir.
- K-means, hierarchical clustering: convex veya küre biçimli kümeler oluşturmaktadır. Hataya neden olabilir.





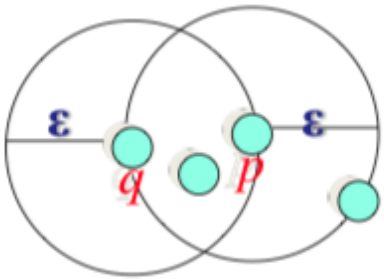


DBSCAN II

Density-based spatial clustering of applications with noise

Algoritma

- İki parametre: ϵ: *epsilon* ve *minimum points* ("MinPts")
- ϵ-Neighborhood bir nesneden ϵ uzaklığında yer alan nesneler. $N_\epsilon(p) = \{q | d(p,q) \leq \epsilon\}$
  - Yüksek yoğunluk (high density): bir nesnenin ϵ-komşuluğunda yer alan nesnelerin sayısı en az **MinPts** olması



ε-Neighborhood of p

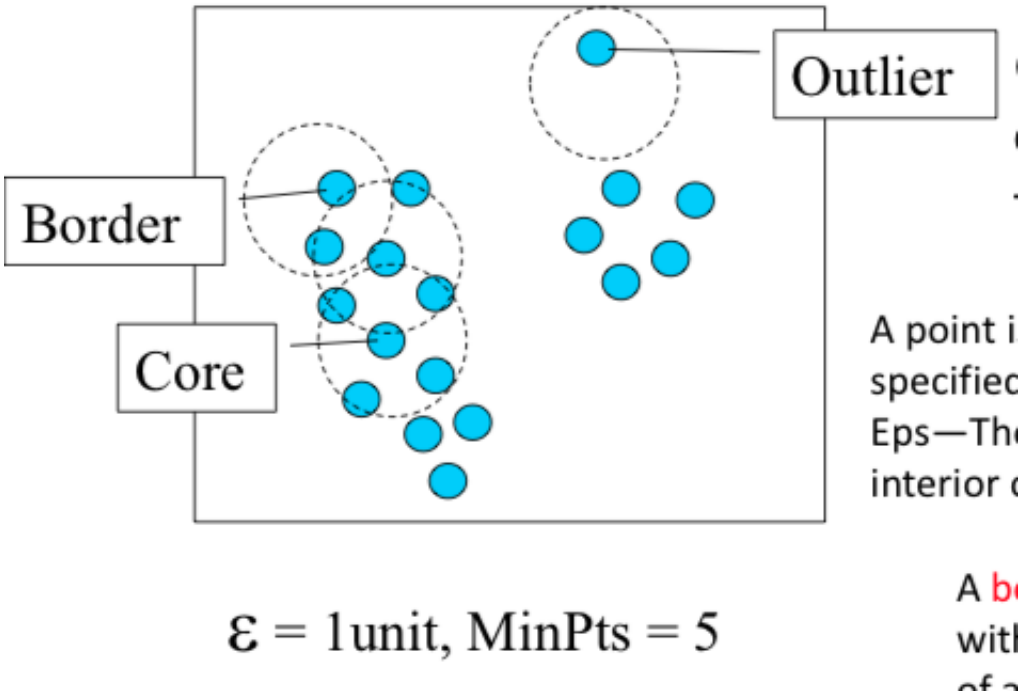
ε-Neighborhood of q

Density of p is "high" (MinPts = 4)

Density of q is "low" (MinPts = 4)

DBSCAN III

Density-based spatial clustering of applications with noise



- ϵ ve **MinPts** verildiğinde nesneler 3 gruba ayrılırlar.
- **Core point**: ϵ içinde **MinPts**'den daha fazla nesne içeren noktalar. **Küme içinde yer alan noktalar**
- **Border point**: MinPts'den daha az nokta içeren fakat bir core point'e komşu.
- **Noise point**: Core point veya border point olmayan noktalar

Python

```
class sklearn.cluster.DBSCAN(eps=0.5,
min_samples=5,metric='euclidean',metric_params=None,
algorithm='auto',leaf_size=30, p=None, n_jobs=1)
```

Model Değerlendirme I

Homogenity

- C: class set, K: cluster set

$$h = 1 - \frac{H(C|K)}{H(C)} \tag{7}$$

- **Python**: sklearn.metrics.homogeneity_score
- Her bir küme içinde yer alan sınıf dağılımı aynı ise $H(C|K) = 0$
- $h = [0, 1]$. 1'e yaklaştıkca doğru kümelenmiştir.

Completeness

$$h = 1 - \frac{H(K|C)}{H(K)} \tag{8}$$

- **Python**: sklearn.metrics.completeness_score
- $h = [0, 1]$. 1'e yaklaştıkca doğru kümelenmiştir.