# Hafta 03/04 - Uzaklık/Benzerlik - En Yakın Komşular - Karar Ağaçları

**BGM 565 - Siber Güvenlik için Makine Öğrenme Yöntemleri**

Bilgi Güvenliği Mühendisliği
Yüksek Lisans Programı

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

İstanbul Şehir Üniversitesi
2018 - Bahar

# İçindekiler

# İçindekiler

# Regularization - Detay I

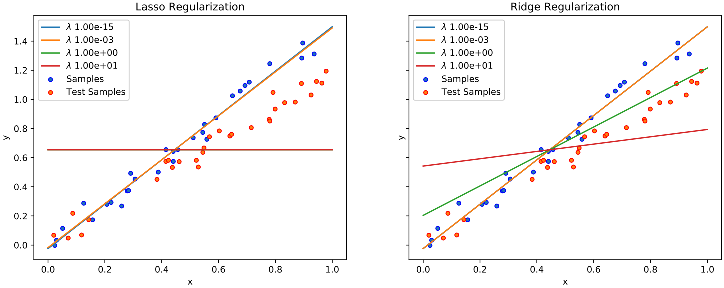
## Regularization

Ridge ve Lasso regresyonu, genellikle yüksek sayıda özniteliğin (feature, column v.s.) olduğu durumlarda model (hipotez) oluşturulmasında kullanılan güçlü tekniklerdir. Burada yüksek, genellikle iki konudan birini ifade etmektedir:

- Bir modelin aşırı öğrenme eğilimini arttırmak için yeterince yüksek sayıda (10 değişken bile aşırı öğrenmeye neden olabilir)
- Hesaplama zorluklarına neden olacak kadar büyük. Modern sistemlerde, bu durum milyonlarca veya milyarlarca özellikte ortaya çıkabilir.

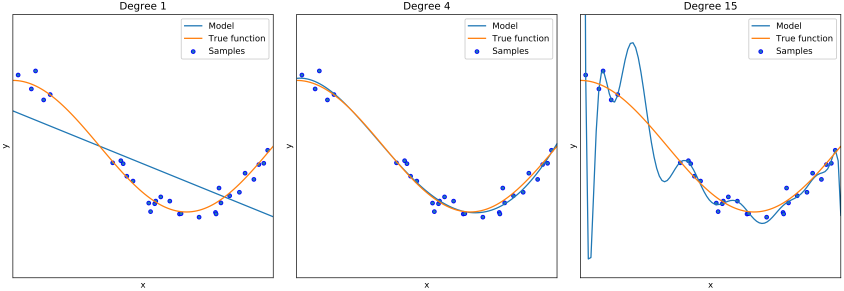
**Regularization**: Tahmin ve gerçek gözlemler arasındaki hatayı en aza indirgeyerek, özellik katsayılarının büyüklüğünü cezalandırarak çalışırlar.

# Regularization - Detay II




**Şekil**: Farklı $\lambda$ değerleri için lasso ve ridge regularization

# Regularization - Detay III




**Şekil**: Aşırı Öğrenme

# Neden Katsayıların Büyüklüğünü Cezalandırıyoruz?

## Regularization

- Eski maliyet fonksiyonu $C = \frac{1}{2m} \sum_{i=1}^{m} \left( h(x^{(i)}) - y^{(i)} \right)^2$
  - MSE (Mean Square Error) değerinin azaltılması tek hedeftir.
  - Bu nedenle, MSE düşürmek için ağırlık değerleri herhangi bir değer alabilir.
  - Ağırlıkların yüksek olması ezberlemeye (overfit) neden olabilir.
- **Çözüm**: Sadece MSE düşürülmemeli aynı zamanda ağırlıklar çok yüksek olmamalıdır.
- Yeni maliyet fonksiyonu $C = \frac{1}{2m} \left( \sum_{i=1}^{m} \left( h(x^{(i)}) - y^{(i)} \right)^2 + \lambda \sum_{j=1}^{n} w_j^2 \right)$
  - $\lambda$: regularization ifadesinin değerini ayarlamak için kullanılmaktadır. $n$ değeri öznitelik sayısıdır.
    - $\lambda$ değeri çok düşükse, maliyet sadece çoğunlukla MSE'ye bağımlıdır.
    - $\lambda$ değeri çok büyükse ağırlıklar oldukça düşük değerler olacaktır.

# Lab

Lab Uygulaması 1 ve 2

# İçindekiler

# Uzaklık I

- Kümeleme ve sınıflandırma olmak üzere iki örneğin birbirlerine olan uzaklık ve benzerlikleri kullanılmaktadır.

## Uzaklık

Uzaklık: $X$ kümesindeki bir metrik. (**uzaklık fonksiyonu** veya sadece **uzaklık** adlandırılır).

$$d : X \times X \rightarrow \mathbb{R} \rightarrow [0, \infty)$$

$\mathbf{x}, \mathbf{y}, \mathbf{z} \in X$ gibi $X$ kümesinden alınacak örnekler için uzaklık olarak $[0, \infty)$ şeklinde skaler sonuçlar üretir. Aşağıdaki koşullar yerine getirir:

- $d(\mathbf{x}, \mathbf{y}) \geq 0$, iki örneklem arasında uzaklık pozitiftir.
- $d(\mathbf{x}, \mathbf{y}) = 0 \Leftrightarrow \mathbf{x} = \mathbf{y}$ iki örneklem arasında uzaklık 0 ise bu iki örneklem aynıdır.
- $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$, Uzaklıklar simetriktir.
- $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{x}, \mathbf{z})$ Üçgen eşitsizliği

Alternatif olarak uzaklık $d$, norm $N$ olarak ifade edebiliriz.

$$d(\mathbf{x}, \mathbf{y}) = N(\mathbf{x} - \mathbf{y})$$

# Uzaklık II

## Uzaklık Matrisi

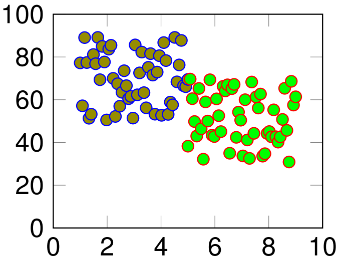
$m$ adet satır ve $n$ adet özniteliğe sahip veri kümesi $\mathcal{D} \in \mathbb{R}^{m \times n}$ için uzaklık matrisi $\mathcal{R} \in \mathbb{R}^{m \times m}$ şeklinde tanımlanır.

$$\mathcal{R} = \begin{bmatrix} 0 & d_{12} & \dots & d_{1m} \\ d_{21} & 0 & \dots & d_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ d_{m1} & d_{m2} & \dots & 0 \end{bmatrix}$$

# Uzaklık III

## Uygulama

Örnekler (satırlar, $\mathbf{x}_i \in \mathbb{R}^n, n \geq 1$) arasında benzerliği birbirlerine olan yakınlık olarak tanımlayabiliriz.



# Farklı Uzaklık Ölçümleri I

## Uzaklık Ölçümleri - Minkowski Uzaklığı

$$d_p(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}||_p = \left( \sum_{k=1}^{n} |\mathbf{x}_k - \mathbf{y}_k|^p \right)^{\frac{1}{p}} \quad (1)$$

- $p = 1$: **Manhattan** uzaklığı ($L_1$-Norm, Taxicab veya City-Block)
- $p = 2$: **Öklid** uzaklığı ($L_2$-Norm veya Ruler)
- $p \to \infty$: **Chebyshev** uzaklığı ($L_{max}$-Norm, maximum metric)
  - $d_{Chebyshev}(\mathbf{x}, \mathbf{y}) = \max_i |\mathbf{x}_i - \mathbf{y}_i|$
  - Örnek:

$$\mathbf{x} = [0, 3, 4, 5]$$
$$\mathbf{y} = [7, 6, 3, -1]$$
$$d_{Chebyshev}(\mathbf{x}, \mathbf{y}) = 7$$

# Benzerlik Ölçümleri I

## Pearson Correlation

### Tanım

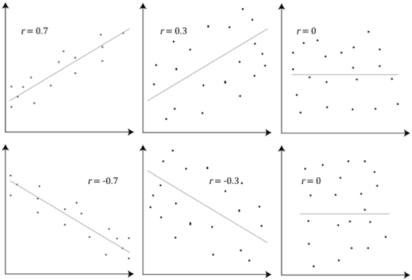
- Pearson korelasyon katsayısı, $X$ ve $Y$ iki değişkeni arasındaki doğrusal korelasyonun bir ölçütüdür.
- Öznitelikler arasında bulunan benzerlik derecesini hesaplamada kullanılır.
- $[-1, +1]$ aralığında yer alırlar.

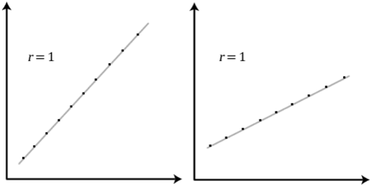
### Pearson Correlation

$$\rho_{X,Y} = \frac{\sum_{k=1}^{n}(x_k - \overline{x})(y_k - \overline{y})}{\sqrt{\sum_{k=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{k=1}^{n}(y_i - \overline{y})^2}} \tag{2}$$

# Benzerlik Ölçümleri II

## Pearson Correlation



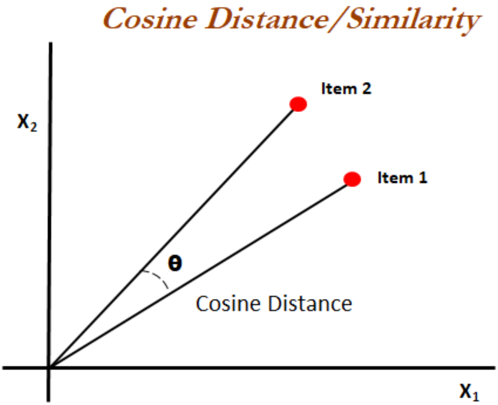
# Kosinüs Benzerliği I

## Kosinüs Benzerliği

- İki vektör arasında bulunan açı farkını kullanan benzerlik ölçümü.
- Açı 0 ise kosinüs değeri 1 olacaktır. Diğer durumlarda 1'den küçük olacaktır.
- Kullanım alanı:
  - Döküman benzerlikleri
  - Multimedia kıyaslamaları
  - Arama motorları v.s.

$$cos(\theta) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \, \|\mathbf{b}\|} = \frac{\sum_{i=1}^{n} \mathbf{a}_i \mathbf{b}_i}{\sqrt{\sum_{i=1}^{n} \mathbf{a}_i^2} \sqrt{\sum_{i=1}^{n} \mathbf{b}_i^2}} \tag{3}$$

# Kosinüs Benzerliği II



# Kosinüs Benzerliği III

## Örnek

$$\mathbf{a} = [1, 6] \;\; \mathbf{b} = [3, 5]$$

$$\|\mathbf{a}\| = \sqrt{1^2 + 6^2} = 6.08 \;\; \|\mathbf{b}\| = \sqrt{3^2 + 5^2} = 5.83$$

$$\mathbf{a} \cdot \mathbf{b} = 1 \times 3 + 6 \times 5 = 33$$

$$cos(\theta) = \frac{33}{6.08 \times 5.83} = 0.93$$

# Kategorik Değişkenler I

## Kategorik Değişkenler

- Öznitelikerin temsili amacıyla söz konusu özelliğin varlığı/yokluğu (Var/Yok, True/False) ile ifade edilmesi

## Hamming Uzaklığı

- *Tanım*: $\mathbf{x} \in \mathbb{R}^m$ $\mathbf{y} \in \mathbb{R}^m$ olsun. $d(\mathbf{x}, \mathbf{y})$ için

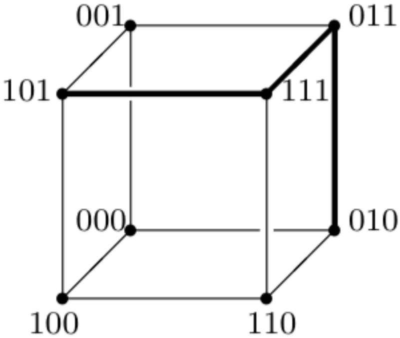
$$d_{hamming}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{m} (\mathbf{x}_i XOR\, \mathbf{y}_i) \tag{4}$$

- Örnekler:

$$d(00111, 11001) = 4 \quad d(0122, 1220) = 3$$

$$d('ka\mathbf{rol}in', 'ka\mathbf{thr}in') = 3$$

# Kategorik Değişkenler II




## İkili alfabe Hamming uzaklığı

Kelimeler ($\mathbb{R}^3$) üç boyutlu bir küpün köşeleri olarak temsil edilebilir.

$$000, 001, 010, 011, 100101, 110, 111$$

# Kategorik Değişkenler III

## Jaccard Benzerliği

- Jaccard benzerliği, hangi üyelerin paylaşıldığını, hangilerinin farklı olduğunu görmek için iki kümeyi karşılaştırmaktadır.
- Benzerlik ölçüsü aralığı: [0, 1]
- **Jaccard = (the number in both sets) / (the number in either set) * 100**
- $J(X, Y) = \frac{|X \cap Y|}{|X \cup Y|}$
- Örnek:

$$
\begin{aligned}
A &= \{0, 1, 2, 5, 6\} \\
B &= \{0, 2, 3, 4, 5, 7, 9\} \\
J(A, B) &= \frac{|A \cap B|}{|A \cup B|} = \frac{|\{0, 2, 5\}|}{|\{0, 1, 2, 3, 4, 5, 6, 7, 9\}|} = \frac{3}{9} = 0.33
\end{aligned}
$$

- **Jaccard uzaklığı**: $d_{jaccard}(A, B) = 1 - J(A, B)$

## Lab

# Lab Uygulaması 3

# İçindekiler

# kNN Yöntemi

## Giriş

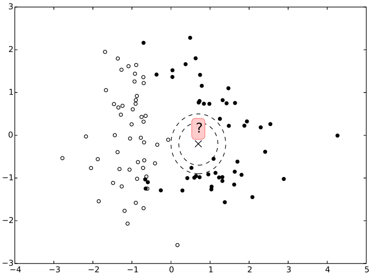
- İki değişkenin benzerliği ve uzaklığı çeşitli yöntemlerle (Öklid, Hamming, Jaccard, Pearson gibi) hesaplanabilmektedir.
- Bu yöntemler kullanılarak **Voronoi** mozaikleriyle **karar sınırları** tanımlanabilmektedir.
- **Voronoi**: Düzlemin belirli bir alt kümesindeki noktalara olan **uzaklığa bağlı olarak** bir düzlemin bölgelere ayrılmasıdır.

# k-En Yakın Komşu Yöntemi I

## k-Nearest-Neighbor (kNN)

### kNN

- En basit sınıflandırma algoritmalarından bir tanesidir.
- Bir test örneği ile belirtilen eğitim örnekleri arasındaki uzaklığa dayanır.




- k-En-Yakın komşu algoritması
- En basit sınıflandırma yöntemi
- **Örnek tabanlı**: Bir öğrenme aşaması yok. Bunun yerine bütün örnekleri hatırlayarak yeni örneğin sınıfını uzaklıkları kullanarak hesaplamaktadır.

# k-En Yakın Komşu Yöntemi II

## k-Nearest-Neighbor (kNN)

### kNN Nasıl Çalışır?

- $k$ adet en yakın komşunun sahip olduğu sınıf etiketi kullanılarak gözlemin sınıfı belirlenir.
- Uzaklık ölçümü ($d(\mathbf{x}, \mathbf{y})$) için genellikle Öklid (Euclidean) uzaklığı kullanılır.
- En yakın komşu:
  - eğer bilinmeyen örnek $\mathbf{x} \in \mathbb{R}^m$ için eğitim veri kümesinde $\mathcal{D} \in \mathbb{R}^{m \times n}$ bulunan örneklerin (satırların) uzaklıklarına göre artan sırada sıralayalım.

$$d_1(\mathbf{x}) \leq d_2(\mathbf{x}) \leq \cdots \leq d_m(\mathbf{x})$$

  - $d_1$ ifadesi en yakın örneğe olan uzaklık, $d_2$ ifadesi ikinci en yakın örneğe olan uzaklığı ifade etmektedir.

# k-En Yakın Komşu Yöntemi III

## k-Nearest-Neighbor (kNN)

### Formal Tanım

- positif tamsayı $K$, gözlemlenen örnek $\mathbf{x}$ ve benzerlik metriği $d$ olmak üzere, kNN aşağıda yer alan 2 adımı gerçekleştirir.
  - Bütün veri kümesi, $\mathcal{D}$ , üzerinde bulunan örnekler (satırlar, gözlemler v.b.) ile sınıf etiketi bilinmeyen $\mathbf{x}$ arasında $d$ hesaplanır.
  - Her bir sınıf için şartlı olasılık hesaplanarak, en yüksek olasılığa sahip sınıf etiketi $\mathbf{x}$ gözlemine atanmaktadır.

$$P(y = j | X = \mathbf{x}) = \frac{1}{K} \sum_{i \in \mathcal{A}} I(y^{(i)} = j)$$

$I$: Indicator function, doğru ise 1 diğer durumda 0.

- kNN'yi anlamak için alternatif bir yol: yeni noktaları sınıflandırmak için kullanılan bir karar sınırını hesaplamak olarak düşünülebilir.

# k-En Yakın Komşu Yöntemi IV

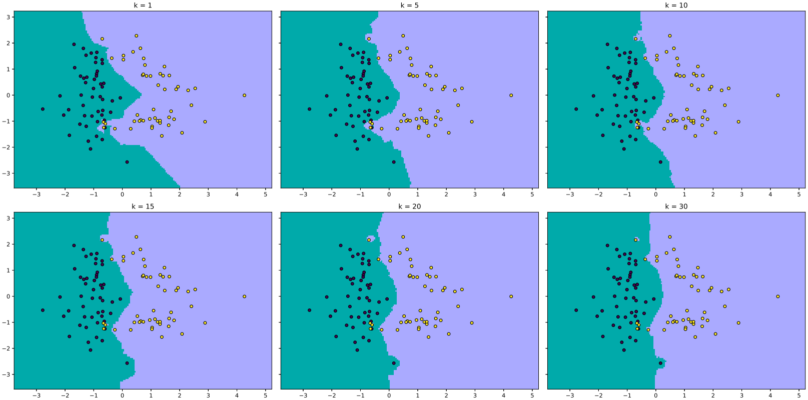
## k-Nearest-Neighbor (kNN)

Table: KDDCUP'99 veri kümesinden alınmış 10 örneğin birbirlerine olan öklid uzaklıkları.

| | $x_0$ | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ | $x_8$ | $x_9$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $x_0$ | 0.00 | 1314.42 | 0.00 | 1.41 | 849.42 | 0.00 | 2022.72 | 1172.66 | 1188.98 | 9325.94 |
| $x_1$ | 1314.42 | 0.00 | 1314.42 | 1313.68 | 842.25 | 1314.42 | 960.09 | 268.17 | 871.97 | 8500.60 |
| $x_2$ | 0.00 | 1314.42 | 0.00 | 1.41 | 849.42 | 0.00 | 2022.72 | 1172.66 | 1188.98 | 9325.94 |
| $x_3$ | 1.41 | 1313.68 | 1.41 | 0.00 | 848.22 | 1.41 | 2022.23 | 1171.83 | 1188.38 | 9325.83 |
| $x_4$ | 849.42 | 842.25 | 849.42 | 848.22 | 0.00 | 849.42 | 1564.92 | 738.86 | 1031.45 | 8962.28 |
| $x_5$ | 0.00 | 1314.42 | 0.00 | 1.41 | 849.42 | 0.00 | 2022.72 | 1172.66 | 1188.98 | 9325.94 |
| $x_6$ | 2022.72 | 960.09 | 2022.72 | 2022.23 | 1564.92 | 2022.72 | 0.00 | 1228.09 | 1775.06 | 7540.70 |
| $x_7$ | 1172.66 | 268.17 | 1172.66 | 1171.83 | 738.86 | 1172.66 | 1228.09 | 0.00 | 649.59 | 8768.61 |
| $x_8$ | 1188.98 | 871.97 | 1188.98 | 1188.38 | 1031.45 | 1188.98 | 1775.06 | 649.59 | 0.00 | 9272.38 |
| $x_9$ | 9325.94 | 8500.60 | 9325.94 | 9325.83 | 8962.28 | 9325.94 | 7540.70 | 8768.61 | 9272.38 | 0.00 |

# Karar Sınırları I

## Decision Boundaries




**Şekil**: kNN karar sınırları

# Karar Sınırları II
## Decision Boundaries

Lab Uygulaması - 4

# Ağırlıklı kNN

Weighted *k*-Nearest-Neighbors

## Ağırlıklı kNN

- **Yöntem**: sınıf etiketi bilinmeyen örneklem **x** için yakınında bulunan örneklerin uzaklıklarına göre ağırlıklandırılması yöntemidir.
- Bu amaçla uzaklık bir **benzerlik** ölçüsü olarak kullanılır. Yakın olan noktaların ağırlıkları daha yüksek olmalıdır.

$$P(y = j | X = \mathbf{x}) = \frac{1}{K} \sum_{i \in \mathcal{A}} \frac{1}{d(\mathbf{x}, \mathbf{x}_i)} I(y^{(i)} = j)$$

## Python

```
class sklearn.neighbors.KNeighborsClassifier(n_neighbors=5,
weights='uniform', algorithm='auto', leaf_size=30, p=2,
 metric='minkowski', metric_params=None,n_jobs=1, **kwargs)
```

# İçindekiler

# Karar Ağaçları I

## Hiyerarşik Öğrenme

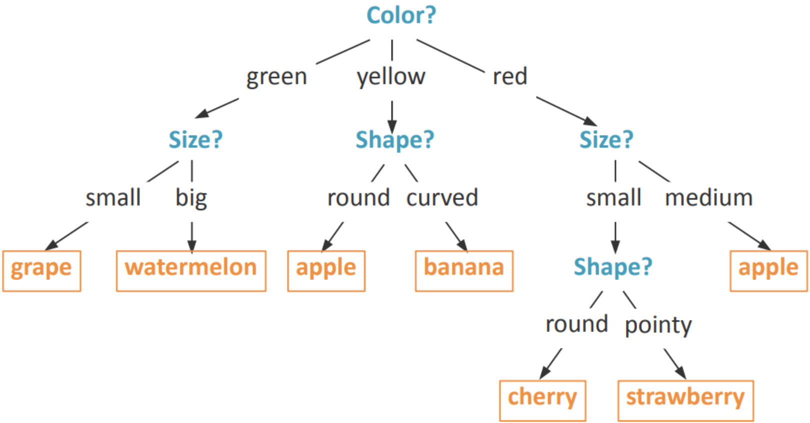
- **Tek Aşamalı Sınıflandırıcılar**
  - Tek bir işlem kullanarak (doğrusal regresyon, lojistik regresyon) bir **x** örneklemine sınıf etiketi ataması yaparlar.
  - Bütün sınıflar için tek bir öznitelik kümesi kullanılmaktadır.
  - Özniteliklerin nominal olma durumu.
- **Hiyerarşik Sınıflandırıcılar**
  - Birden çok ardışık test

# Karar Ağaçları II

## Nominal Veri

- **Öznitelikler**
  - Ayrık (Discrete)
  - Herhangi bir **sıralama/benzerlik** kavramı yok
  - Metrik olmayan öğrenme (Non-metric learning)
- **Örnekler**:
  - **Protocol type**: TCP, UDP
  - **Service**: http, mail, ftp, ssh
  - **Flags**: SYN, ACK, RST, FIN

# Karar Ağaçları III



# Çok Sınıflı Sınıflandırma
Multiclass classification

## Çok Sınıflı Sınıflandırma

- **Bire-karşı-hepsi (One-versus-all)**
  - $K$ adet sınıflandırıcı (hipotez, model) oluştur, oylama ile **x** etikietini bul
- **Bire-karşı-bir (One-versus-one)**:
  - K(K-1)/2 adet sınıflandırıcı (hipotez, model) oluştur, oylama ile **x** etikietini bul
- Çoklu sınıfları işleyen bir algoritma kullanımı:
  - Karar ağaçları
  - Yapay Sinir ağları

# Ağacın Oluşturulması I

## Giriş

- Monothetic ağaçlarda (her bir node üzerinde sadece bir öznitelik olması) karar sınırları axis'lere ortagonal yapıdadır.
- Ayrıştırma değişkeni (j) ve ayrıştırma noktaları 2 bölge tanımlamaktadır.
- *j* ve *s* değerleri saf olmamayı (Impurity) azaltacak şekilde seçilmelidir.
  - Niteliklerin kategorik olması durumunda: **Bilgi Kazanımı**
  - Niteliklerin sürekli olması durumunda: **Gini index**

## Formal Tanım

- Eğitim vektörleri $\mathbf{x}_i \in \mathbb{R}^n, i = 1, \cdot, m$ ve sınıf vektörü $\mathbf{y} \in \mathbb{R}^m$ olsun.
- Karar ağacı algoritması, girdi uzayını aynı etiketlere sahip örneklere göre parçalamaktadır.
- **Amaç**: Impurity minimize etmek için öznitelikler seçilir.

$$\min_j \left( \frac{|R_1(j)|}{n_{total}} \times Imp(R_1(j)) + \frac{|R_2(j)|}{n_{total}} \times Imp(R_2(j)) \right) \tag{5}$$

# Ağacın Oluşturulması II

## Bilgi Kazanımı (Information Gain)

- Rassal bir değişken $X$ belirsizliği/rassallığı **Entropi** ile ölçülür.

$$H(X_n) = -\sum_k p_k log_2(p_k) \quad (6)$$

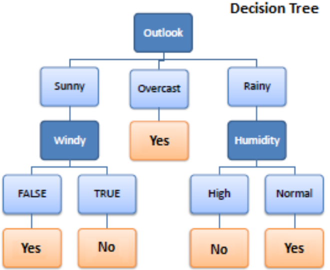
  $p_k$ ifadesi, $X_n$ içinde yer alan eğitim örneklerinin $k$ sınıfına ait olanlarının oranıdır.
- İkili sınıflandırma problemi (-1, +1):
  - Bütün örneklerin sınıf etiketi aynı ise Entropi 0. Düşük Entropi.

    $$\mathbf{y} = \{-1, -1, \cdots, -1\} \, veya \, \mathbf{y} = \{+1, +1, \cdots, +1\}$$

  - Bütün örneklerin yarısının sınıf etiketi $-1$ diğer yarısı $+1$ aynı ise Entropi 1. Yüksek Entropi

# Ağacın Oluşturulması III

| Predictors | | | | Target |
|---|---|---|---|---|
| Outlook | Temp. | Humidity | Windy | Play Golf |
| Rainy | Hot | High | False | No |
| Rainy | Hot | High | True | No |
| Overcast | Hot | High | False | Yes |
| Sunny | Mild | High | False | Yes |
| Sunny | Cool | Normal | False | Yes |
| Sunny | Cool | Normal | True | No |
| Overcast | Cool | Normal | True | Yes |
| Rainy | Mild | High | False | No |
| Rainy | Cool | Normal | False | Yes |
| Sunny | Mild | Normal | False | Yes |
| Rainy | Mild | Normal | True | Yes |
| Overcast | Mild | High | True | Yes |
| Overcast | Hot | Normal | False | Yes |
| Sunny | Mild | High | True | No |




**Şekil**: Örnek verikümesi [1]

# Ağacın Oluşturulması IV

## Bağımlı Değişken Entropy Değeri

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

| Play Golf | |
|---|---|
| Yes | No |
| 9 | 5 |

Entropy(PlayGolf) = Entropy (5,9)

= Entropy (0.36, 0.64)

= - (0.36 $\log_2$ 0.36) - (0.64 $\log_2$ 0.64)

= 0.94

# Ağacın Oluşturulması V

## Bağımsız Değişken (Outlook) Entropy Değeri

$$E(T,X) = \sum_{c \in X} P(c)E(c)$$

| | | Play Golf | | |
|---|---|---|---|---|
| | | Yes | No | |
| Outlook | Sunny | 3 | 2 | 5 |
| | Overcast | 4 | 0 | 4 |
| | Rainy | 2 | 3 | 5 |
| | | | | 14 |

**E**(PlayGolf, Outlook) = **P**(Sunny)***E**(3,2) + **P**(Overcast)***E**(4,0) + **P**(Rainy)***E**(2,3)

= (5/14)*0.971 + (4/14)*0.0 + (5/14)*0.971

= 0.693

# Ağacın Oluşturulması VI

## Bilgi Kazanımı (Information Gain)

**Bilgi kazancı**, bir veri kümesinin bir öznitelik üzerine bölünmesinden sonra **Entropi azalmasına** dayanmaktadır.

| | | Play Golf | |
|---|---|---|---|
| | | Yes | No |
| Outlook | Sunny | 3 | 2 |
| | Overcast | 4 | 0 |
| | Rainy | 2 | 3 |
| Gain = 0.247 | | | |

| | | Play Golf | |
|---|---|---|---|
| | | Yes | No |
| Temp. | Hot | 2 | 2 |
| | Mild | 4 | 2 |
| | Cool | 3 | 1 |
| Gain = 0.029 | | | |

| | | Play Golf | |
|---|---|---|---|
| | | Yes | No |
| Humidity | High | 3 | 4 |
| | Normal | 6 | 1 |
| Gain = 0.152 | | | |

| | | Play Golf | |
|---|---|---|---|
| | | Yes | No |
| Windy | False | 6 | 2 |
| | True | 3 | 3 |
| Gain = 0.048 | | | |

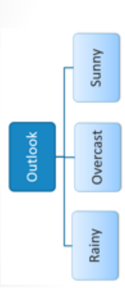
$$Gain(T,X) = Entropy(T) - Entropy(T,X)$$

**G**(PlayGolf, Outlook) = **E**(PlayGolf) – **E**(PlayGolf, Outlook)

= 0.940 – 0.693 = 0.247

# Ağacın Oluşturulması VII

En yüksek bilgi kazanımı olan öznitelik (Outlook), karar düğümü olarak seçilir.




| Outlook | Temp | Humidity | Windy | Play Golf |
|---|---|---|---|---|
| Sunny | Mild | High | FALSE | Yes |
| Sunny | Cool | Normal | FALSE | Yes |
| Sunny | Cool | Normal | TRUE | No |
| Sunny | Mild | Normal | FALSE | Yes |
| Sunny | Mild | High | TRUE | No |
| Overcast | Hot | High | FALSE | Yes |
| Overcast | Cool | Normal | TRUE | Yes |
| Overcast | Mild | High | TRUE | Yes |
| Overcast | Hot | Normal | FALSE | Yes |
| Rainy | Hot | High | FALSE | No |
| Rainy | Hot | High | TRUE | No |
| Rainy | Mild | High | FALSE | No |
| Rainy | Cool | Normal | FALSE | Yes |
| Rainy | Mild | Normal | TRUE | Yes |

# Ağacın Oluşturulması VIII

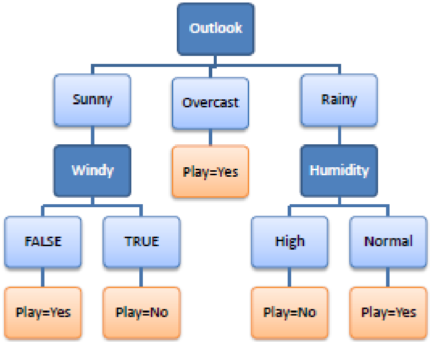
$R_1$: IF (Outlook=Sunny) AND (Windy=FALSE) THEN Play=Yes

$R_2$: IF (Outlook=Sunny) AND (Windy=TRUE) THEN Play=No

$R_3$: IF (Outlook=Overcast) THEN Play=Yes

$R_4$: IF (Outlook=Rainy) AND (Humidity=High) THEN Play=No

$R_5$: IF (Outlook=Rain) AND (Humidity=Normal) THEN Play=Yes




[1] http://www.saedsayad.com/decision_tree.htm