

Karar Ağaçları

Prof. Dr. Hamdi Tolga KAHRAMAN

Karar Ağaçları

- Verinin içerdiği ortak özelliklere göre ayrıştırılması işlemi sınıflandırma olarak adlandırılır.
- Birçok sınıflandırma yöntemi vardır; karar ağaçları bunlardan birisidir.
- Karar ağaçları oluşturmak için temel olarak entropiye dayalı algoritmalar, sınıflandırma ve regrasyon ağaçları, bellek tabanlı sınıflandırma modelleri biçiminde birçok yöntem geliştirilmiştir.
- ID3 ve C4.5 algoritmaları entropiye dayalı karar ağacı oluşturma yöntemleridir.

Karar Ağaçları

- Karar ağacı öğrenme, ayrık (discrete) değerli fonksiyonlara yakınsamak için kullanılan bir metottur. Öğrenilen fonksiyon bir karar ağacı olarak ifade edilir.
- Sınıflandırma problemleri için yaygın kullanılan yöntemdir.
- Sınıflandırma doğruluğu diğer öğrenme metotlarına göre çok etkindir.
- Bu teknikte sınıflandırma için bir ağaç oluşturulur ve daha sonra yeni durumlar bu ağaca uygulanarak sınıflandırılır.
- **ID₃ ve C_{4.5}, entropiye dayalı** sınıflandırma algoritmalarıdır.

Karar Ağaçları

- Örneklerde oluşan bir küme kullanılarak karar ağacının oluşturulmasını sağlayan çok sayıda öğrenme yöntemi vardır.
- Bu teknik en iyi tahmine ulaşmak için bağımlı ve bağımsız değişkenler arasındaki olası tüm ilişkilerin araştırılmasına dayanmaktadır.
- Karar ağacı tekniğinde en kuvvetli ilişkiye sahip bağımsız değişken bulunduğunda veri kümesi bu bağımsız değişkenin değerlerine göre ikiye ayrılmaktadır. Bu süreç olası bölünmeler tamamlanıncaya kadar devam etmektedir.

Karar Ağaçları

- Karar ağaçları akış şemalarına benzeyen yapılardır. Karar ağaçları karar düğümleri, dallar ve yapraklardan meydana gelir.
 - a. Her bir nitelik bir 'düğüm' tarafından temsil edilir. 'Dallar' ve 'yapraklar' ağaç yapısının elemanlarıdır. En son yapı 'yaprak' ve en üst yapı 'kök' ve bunların arasında kalan yapılar ise 'dal' olarak isimlendirilir.
 - b. Karar düğümü, yapılacak testi belirtir.
 - c. Bu testin sonucunda karar ağacı herhangi bir veri kaybına uğramadan dallara ayrılmaktadır.
 - d. Her düğümde test ve dallara ayrılma işlemi ardışık olarak gerçekleşmektedir. Ayrılma işlemi üst seviyedeki ayrımlara bağımlı olmaktadır.
 - e. Her bir dal sınıflama işlemini tamamlamaya adaydır.

Karar Ağaçları

- Karar ağaçları karar düğümleri, dallar ve yapraklardan meydana gelir.
 - e. Eğer bir dalın ucunda sınıflama işlemi gerçekleşmiyorsa, bu noktada bir karar düğümü meydana gelmektedir.
 - f. Eğer sınıflama işlemi yapılabiliyorsa, bu durum o dalın sonunda yaprak olduğunu ifade etmektedir. İşte bu yaprak veri kümesi üzerinde belirlenmek istenen sınıflardan birisi olmaktadır.
 - g. Kısaca karar ağacı kök düğümden başlayarak, yukarıdan aşağıya doğru ardışık düğümler takip edilerek yaprağa ulaşıncaya kadar devam eden bir süreçtir.

Karar Ağaçları

- Karar ağaçlarında kullanılan farklı algoritmalar bulunmaktadır. bunlardan bazıları C4.5, CHAID, CART, ID₃ (genel olarak birbirlerine çok benzerler)
- Karar ağaçlarında kullanılan temel algoritma yapısı:
 1. Başlangıçta bütün noktalar ağacın kökünde toplanmaktadır
 2. Tüm örneklemeler aynı sınıfa ait olması durumunda, düğüm yaprağa dönüşür ve aynı isim ile adlandırılır.
 3. Aksi halde düğümdeki örneklemeler birden fazla sınıfa aittir. Bu durumda test yapılarak karar verilir ve bir bölümlenme meydana gelir.
 4. Kategorik veriler kullanılmaktadır. Sürekli değişkenlerin kesikli değişken haline dönüştürülmesi gerekmektedir.
 5. Bir dal, test değişkeninin tüm değerleri için oluşturulmakta ve örneklemin bölümlenmesi buna göre yapılmaktadır.

Karar Ağaçları

- Karar ağaçlarında kullanılan temel algoritma yapısı:
- 6. Örneklemin her bölümlenmesinde yinelemeli olarak aynı algoritma kullanılmaktadır.
- 7. Bölümlemenin sona ermesi için aşağıdaki koşullardan birisinin gerçekleşmesi gerekmektedir.
 - a. Bir düğümde bulunan bütün örneklemeler aynı sınıfa aittir.
 - b. Bölümlemenin yapılacağı başka değişken kalmamıştır.
 - c. Başka örneklem kalmamıştır.

Karar Ağaçları Uygulama Alanları

- Kişilerin kredi geçmişlerini kullanarak kredi kararlarının verilmesi
- Geçmişte işletmeye en faydalı olan bireylerin özelliklerini kullanarak işe alma süreçlerinin belirlenmesi
- Tıbbi gözlem verilerinden yararlanarak en etkin kararların verilmesi
- Hangi değişkenlerin satışları etkilediğinin belirlenmesi
- Üretim verilerini inceleyerek ürün hatalarına yol açan değişkenlerin belirlenmesi gibi uygulamalarda kullanılmaktadır.

Karar Ağacı Öğrenmesi

- Karar ağacı öğrenmek, bir öğrenme kümesinden bir ağaç oluşturmak demektir.
- Bir öğrenme kümesini hatasız öğrenen birden çok karar ağacı olabilir
- Basitlik ilkesi nedeniyle bu ağaçların en küçüğü bulunmak istenir.
- Bir ağacın büyüklüğü düğüm sayısına ve bu düğümlerin karmaşıklığına bağlıdır.

Karar Ağaçları

- **Entropi**, rastgele değere sahip bir değişken veya bir sistem için **belirsizlik ölçütüdür.**
- **Enformasyon**, rastsal bir olayın gerçekleşmesi halinde ortaya çıkan bilgi ölçütüdür.
- Bir süreç için entropi, tüm örnekler tarafından içerilen **enformasyonun beklenen değeridir.**
- **Eşit olasılıklı durumlara sahip sistemler yüksek belirsizliğe sahiptirler.**
- Shannon, bir sistemdeki durum değişikliğinde, **entropideki değişimin enformasyon boyutunu tanımladığını öne sürmüştür.**
- Buna göre **bir sistemdeki belirsizlik arttıkça, bir durum gerçekleştiğinde elde edilecek enformasyon boyutu da artacaktır.**

Karar Ağaçları

- Shannon bilgiyi bitlerle ifade ettiği için, logaritmayı 2 tabanında kullanmıştır ve enformasyon formülünü aşağıdaki gibi vermiştir.

$$I(x) = \log \frac{1}{P(x)} = -\log P(x)$$

- $P(x)$, x olayının gerçekleşme olasılığını gösterir.
- Shannon'a göre entropi, **iletile bir mesajın taşıdığı enformasyonun beklenen değeridir.**
- Shannon entropisi H , aşağıdaki gibi ifade edilir:

$$\begin{aligned} H(X) &= E(I(X)) = \sum_{1 \leq i \leq n} P(x_i) \cdot I(x_i) \\ &= \sum_{i=1}^n P(x_i) \log_2 \frac{1}{P(x_i)} = -\sum_{i=1}^n P_i \log_2 P_i \end{aligned}$$

Karar Ağaçları

Örnek

- Bir paranın havaya atılması olayı rastsal X sürecini gösterebiliriz. Yazı ve tura gelme olasılıkları eşit olduğundan elde edilecek enformasyon,

$$I(X) = \log \frac{1}{P(X)} = \log \frac{1}{0,5} = \log 2 = 1$$

olur. Bu olayın sonucunda 1 bitlik bilgi kazanılmıştır.

- Entropi değeri ise 1 olarak bulunur.

$$\begin{aligned} H(X) &= -\sum_{i=1}^2 p_i \log_2 p_i \\ &= -(0.5 \log_2 0.5 + 0.5 \log_2 0.5) = 1 \end{aligned}$$

Karar Ağaçları

Örnek

- Aşağıdaki 8 elemanlı S kümesi verilsin.

$$S = \{evet, hayır, evet, hayır, hayır, hayır, hayır, hayır\}$$

- “*evet*” ve “*hayır*” için olasılık,

$$p(evet) = \frac{2}{8} = 0,25 \quad p(hayir) = \frac{6}{8} = 0,75$$

- Entropi değeri,

$$\begin{aligned} H(S) &= p(evet) \log_2 \frac{1}{p(evet)} + p(hayir) \log_2 \frac{1}{p(hayir)} \\ &= 0,25 \cdot \log_2 \frac{1}{0,25} + 0,75 \cdot \log_2 \frac{1}{0,75} \\ &= 0,81 \end{aligned}$$

ID3

Karar ağacı yaklaşımı, sınıflandırma problemlerinde oldukça faydalıdır. Karar ağaçlarının oluşturulması sırasında dallanmaya/bölümlemeye hangi nitelikten başlanacağı oldukça önemlidir. Zira, sınırlı sayıda kayıttan oluşan bir eğitim kümesinden faydalananarak mümkün tüm ağaç yapılarını ortaya çıkarmak ve içlerinden en uygun olanını seçerek ondan başlamak çok fazla alternatiften dolayı kolay bir iş değildir. O nedenle karar ağacı algoritmalarının çoğu, daha başlangıçta bir takım değerleri hesaplayarak ona göre ağaç oluşturma yoluna gitmektedirler. Bu amaçla entropi kavramı kullanılabilir ve ağacın dallanması entropinin alacağı değere göre gerçekleştirilebilir

ID3 (Iterative Dichotomiser 3), karar ağacı tabanlı bir algoritmadır ve Ross Quinlan tarafından 1979 yılında geliştirilmiştir. Algoritma, bir veri setinden bir karar ağacı oluşturulmasıyla ilgilidir. Algoritmada, hedef sınır değerlerini içeren nitelik belirlenir (bu niteliğin değerleri S kümesidir) ve bu niteliğin kümesi için "Entropi" hesaplanır

ID3

$$Entropi(S) = - \sum_{x \in X} p(x) \log_2 p(x)$$

Burada, X , S 'deki sınıfların kümesini, $p(x)$, x sınıfındaki eleman sayısının S 'deki elemanların sayısına oranıdır. Entropi (S)=0 ise, S kümesi tam olarak sınıflandırılmış demektir.

Karar ağacının hangi nitelikten dallanacağını belirlemek için, hedef için kullanılan entropi değeri kullanılarak her bir diğer niteliğin "Bilgi kazancı ya da Kazanç" hesaplanması

$$Kazanç(S, A) = Entropi(S) - \sum_{v \in A} \frac{|S_v|}{|S|} \cdot Entropi(S_v)$$

Bu kazançlardan en yükseğine sahip olan nitelik, dallanacak nitelik olarak belirlenir. Dallanılan niteliğin her bir sınıfı için dallanma seçenekleri belirlenmeye çalışılır ve karar ağacı oluşturulur.

ID 3

- Diğer tümevarımsal (inductive) öğrenme metotlarında olduğu gibi, ID₃ de bir hipotez uzayındaki en iyi hipotezi arar.
- ID₃ tarafından taranan hipotez uzayı mümkün olan bütün karar ağaçlarıdır.
- ID₃ basitten karmaşığa bir yol izleyerek başta boş bir karar ağacı ile başlayıp gittikçe daha ayrıntılı ve eğitim verisine uygun bir karar ağacına ulaşır.
- Bu aramaya yol gösteren fonksiyon ise bilgi kazanımı ölçüsüdür.

ID 3

- ID₃'ye arama uzayı ve arama stratejisi açısından bakacak olursak, yapabildiklerini ve bazı limitasyonları görebiliriz.
- ID₃'nin arama uzayı tamdır (complete). Her ayrık değerli fonksiyon bir karar ağacı ile gösterilebileceği için, ID₃'nin arama uzayı da bütün karar ağaçları uzayı olduğu için, ID₃'nin hedef fonksiyonu bulamama gibi bir problemi yoktur.
- ID₃ her an tek bir hipotezi hesaba kattığı için birden fazla hipotez çıktısı veremez. Hipotez uzayında aynı fonksiyonu gösterebilen başka hipotezler olup olmadığı sorusuna cevap veremez.

ID 3

- ID3 çalışırken hiç geriye dönüş (backtrack) yapmaz. Bir özelliğe karar verirse örneğin, bunu daha sonra dönüp tekrar kontrol etmez. Bu durumda da bu tip algoritmaların yaşadığı genel sorunla karşılaşabilir : Global olarak değil de lokal olarak bir en iyi sonuca ulaşmak.
- ID3, bütün örnekleri aynı anda kullanan hesaplamalar yapar (bilgi kazanımı hesabı gibi). Bu, FIND-S ve CANDIDATE-ELIMINATION gibi, örnekleri tek tek ele alan algoritmalarından farklıdır. ID3, bu sayede herhangi bir eğitim verisi örneğinde bulunabilecek hatalara karşı daha dirençlidir. Gürültülü (noisy) veri ile de kullanılabilecek şekilde kolayca geliştirilebilir.

Karar Ağaçları

- Örnek; Havanın beyzbol oynamak için elverişli olup olmadığına karar vermek isteniyor. Toplanan iki haftalık veriler Çizelgede verilmektedir. ID3 algoritmasıyla karar ağacı oluşturulması istenmektedir.

Gün	Dışarı	Sıcaklık	Nem	Rüzgar	Oyun
D1	Güneşli	Sıcak	Yüksek	Hafif	Hayır
D2	Güneşli	Sıcak	Yüksek	Kuvvetli	Hayır
D3	Kapalı	Sıcak	Yüksek	Hafif	Evet
D4	Yağmurlu	Ilık	Yüksek	Hafif	Evet
D5	Yağmurlu	Soğuk	Normal	Hafif	Evet
D6	Yağmurlu	Soğuk	Normal	Kuvvetli	Hayır
D7	Kapalı	Soğuk	Normal	Kuvvetli	Evet
D8	Güneşli	Ilık	Yüksek	Hafif	Hayır
D9	Güneşli	Soğuk	Normal	Hafif	Evet
D10	Yağmurlu	Ilık	Normal	Hafif	Evet
D11	Güneşli	Ilık	Normal	Kuvvetli	Evet
D12	Kapalı	Ilık	Yüksek	Kuvvetli	Evet
D13	Kapalı	Sıcak	Normal	Hafif	Evet
D14	Yağmurlu	Ilık	Yüksek	Kuvvetli	Hayır

Hedef sınıf değerlerini içeren "Oyun" dur. Oyun nitelik değerlerinden oluşan küme S kümesidir. Kümede "9 Evet" ve "5 Hayır" bulunmaktadır.

$$Entropi(OYUN) = - \left[\frac{9}{14} \log_2 \frac{9}{14} + \frac{5}{14} \log_2 \frac{5}{14} \right] = 0,940$$

Herbir nitelik için kazanç değerleri hesaplanır;

$$|DIŞARI_{Güneşli}| = 5 \text{ kayıt var} \longrightarrow \text{Karşılık gelen 2 Evet, 3 Hayır}$$

$$|DIŞARI_{Yağmurlu}| = 5 \text{ kayıt var} \longrightarrow \text{Karşılık gelen 3 Evet, 2 Hayır}$$

$$|DIŞARI_{Kapalı}| = 4 \text{ kayıt var} \longrightarrow \text{Karşılık gelen 4 Evet, 0 Hayır}$$

$$Entropi |DIŞARI_{Güneşli}| = - \left[\frac{2}{5} \log_2 \frac{2}{5} + \frac{3}{5} \log_2 \frac{3}{5} \right] = 0,971$$

$$Entropi |DIŞARI_{Yağmurlu}| = - \left[\frac{3}{5} \log_2 \frac{3}{5} + \frac{2}{5} \log_2 \frac{2}{5} \right] = 0,971$$

$$Entropi |DIŞARI_{Kapalı}| = - \left[\frac{4}{4} \log_2 \frac{4}{4} \right] = 0 \text{ olarak bulunur.}$$

$$Kazanç(DIŞARI, OYUN) = 0,940 - \left(\frac{5}{14} (0,971) + \frac{5}{14} (0,971) + \frac{4}{14} (0) \right) = 0,246$$

Benzer şekilde diğer nitelikler için de entropi/kazanç hesaplamaları yapıldığında aşağıdaki tablo değerleri (Çizelge) elde edilecektir.

Çizelge Kazanç değerleri

Nitelik	Kazanç
Dışarı	0,246
Sıcaklık	0,029
Nem	0,151
Rüzgar	0,048

Çizelgeden görüleceği üzere, en büyük kazançlı nitelik "Dışarı" dır. Dolayısıyla dallanma da bu nitelikten olacaktır. Kök düğüm "Dışarı" dır ve 3 dala sahiptir (Güneşli, Kapalı ve Yağmurlu). Şimdiki soru, "Güneşli" dal düğümünde hangi niteliğin test edileceğidir.
 $SGüneşli = \{D_1, D_2, D_8, D_9, D_{11}\}$.

- Yeni tablo, sadece 5 satıra karşılık gelen değerlerden oluşacaktır.

Çizelge. Dışarı=Güneşli için veriler

Gün	Dışarı	Sıcaklık	Nem	Rüzgar	Oyun
D1	Güneşli	Sıcak	Yüksek	Hafif	Hayır
D2	Güneşli	Sıcak	Yüksek	Kuvvetli	Hayır
D8	Güneşli	Ilık	Yüksek	Hafif	Hayır
D9	Güneşli	Soğuk	Normal	Hafif	Evet
D11	Güneşli	Ilık	Normal	Kuvvetli	Evet

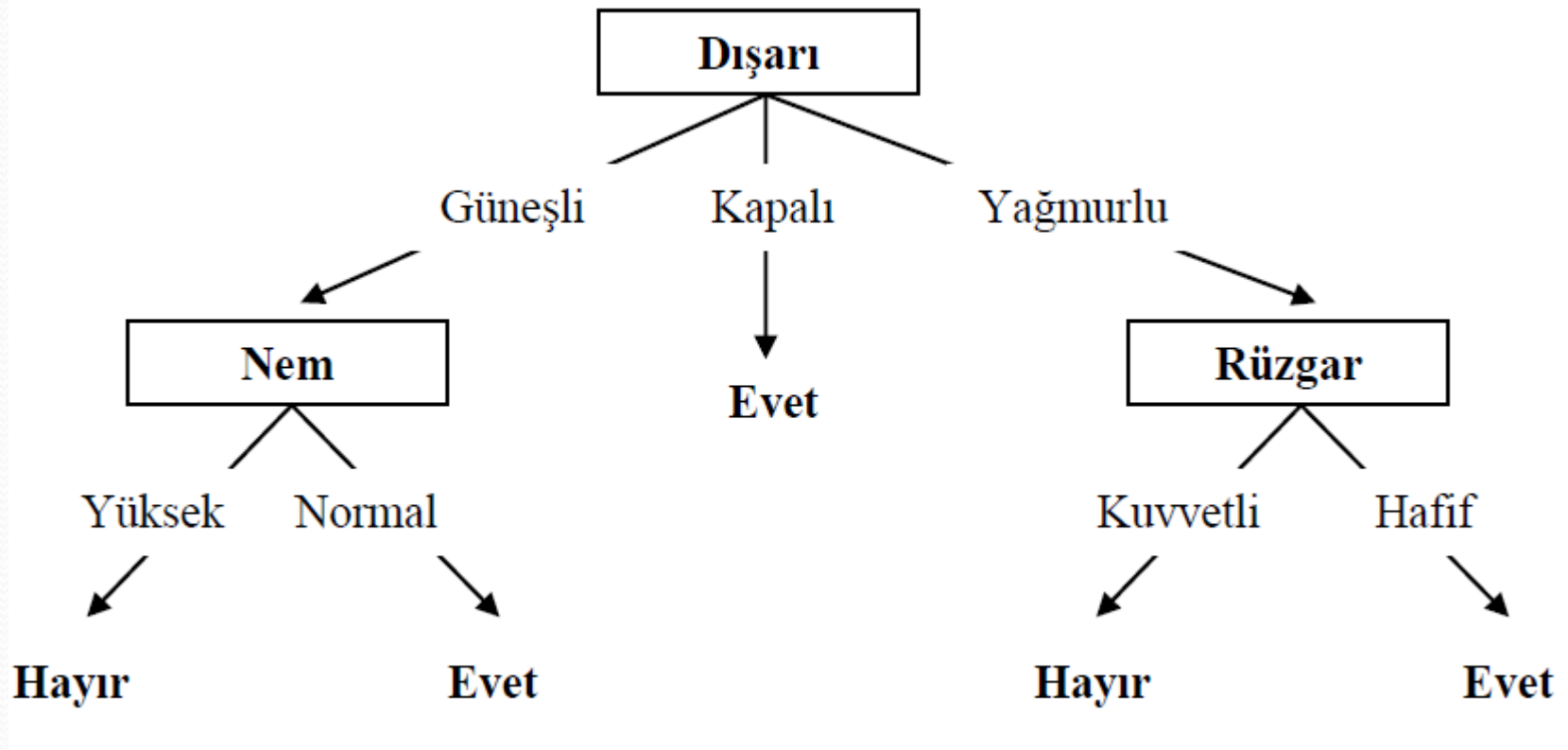
Burada önce "Oyun" için yine entropi hesaplanır ve her bir nitelik için önceden olduğu gibi kazanç değerleri hesaplanır.

- Benzer hesaplamalar tekrarlandığında Çizelgedeki değerlere ulaşılabilir:

Nitelik	Kazanç
Sıcaklık	0,570
Nem	0,970
Rüzgar	0,019

- Çizelgeden görüleceği üzere, en büyük kazançlı nitelik "Nem" dir. Dolayısıyla dallanma da bu nitelikten olacaktır. Süreç, tüm veriler sınıflandırılana kadar devam edecektir.

- Algoritma sonucu elde edilen karar ağacı Şekilde verilmektedir.



ID3 Nihai karar ağacı

Karar ağacı kural formatında da ifade edilebilir (IF Dışarı=Güneşli AND Nem=Yüksek THEN Oyun=Hayır vb.)

C4.5 ve C5.0 Karar ağacı algoritması

- C4.5 karar ağacı algoritması, ID3'ün geliştirilmiş hali, üst versiyonudur.
- Algoritma, en iyi bilinen ve en yaygın kullanılan öğrenme algoritmalarından birisidir.
- Karar ağacı oluşturulmasında ID3 algoritması, "Bilgi kazancı" ölçüsünü kullanırken, C4.5 ise, "Kazanç Oranı" ölçüsünü kullanmaktadır.
- C4.5 algoritması, ID3'e ilaveten (ID3 kısıtlamalarını kaldırmak için) bazı ilave durumlar içermektedir:
 - a) Sayısal (kesikli ve sürekli) değerli nitelikleri de izin verir.
 - b) Bilinmeyen (eksik) değerleri kümelerin olmasına izin verir.
 - c) Gereksiz alt ağaçları budayarak ağacın yapısını basitleştirir.

Sürekli (Continuous) Değerli Özellikler

ID3'nin yukarıda verilen tanımı, özelliklerin ayrık değerler aldığını varsayar. Hem hedef kavram, hem de her düğümde test edilen özellikler ayrık değerli olmalıdır. Özellikler üzerindeki kısıt, kolayca kaldırılabilir. Bu sürekli değerlere sahip bir özelliği ayrık değerler kümesine ayırmakla yapılabilir. Yani, herhangi bir sürekli değerli A özelliği için, algoritma dinamik olarak A_c gibi, $A < c$ iken doğru değilken yanlış olan, Boole değerli bir özellik tanımlayabilir. Buradaki tek soru c eşik değerinin nasıl seçileceğidir.

Örnek olarak, Sıcaklık özelliğinin sürekli haline bakalım. Sıcaklık özelliği ve *TenisOyna* hedef kavramının aşağıdaki gibi değerler aldığını varsayalım.

<i>Sıcaklık</i>	40	48	60	72	80	90
<i>TenisOyna</i>	Hayır	Hayır	Evet	Evet	Evet	Hayır

Burada c 'yi bilgi kazanımını maksimum yapacak şekilde seçmemiz gerektiği açıktır. Örnekleri sürekli değerli özellik A 'ya göre sıralayıp, daha sonra yan yana olan hangi değerlerde hedef kavramın değiştiğini bulursak, bu değerlerin ortalaması ile bir küme aday eşik değer oluşturabiliriz. Bu aday eşik değerleri kullanarak her biri için bilgi kazanımını hesaplayabiliriz. Yukarıdaki örnekte, bu prosedüre göre, iki aday eşik değer vardır: $(48+60)/2$ ve $(80+90)/2$. Daha sonra bu iki özellik ($Sıcaklık_{54}$ ve $Sıcaklık_{85}$) için de bilgi kazanımı hesaplanırsa en iyi özellik bulunacaktır (bu $Sıcaklık_{54}$ 'tür). Bundan sonra bulunan en iyi özellik aynen diğer özellikler gibi ID3 algoritmasında kullanılabilir.

Ağaç Budama

Budama, sınıflandırmaya katkısı olmayan bölümlerin karar ağacından çıkarılması işlemidir.

Bu sayede karar ağacı hem sade hem de anlaşılabilir hale gelir. İki çeşit budama yöntemi vardır;

- ön budama
- sonradan budama

Ön Budama

Ön budama işlemi ağaç oluşturulurken yapılır. Bölünen nitelikler, değerleri belli bir eşik değerinin (hata toleransının) üstünde değilse o noktada ağaç bölümlleme işlemi durdurulur ve o an elde bulunan kümedeki baskın sınıf etiketi, yaprak olarak oluşturulur.


Sonradan Budama


Sonradan budama işlemi ağaç oluşturulduktan sonra devreye girer. Alt ağaçları silerek yaprak oluşturma, alt ağaçları yükseltme, dal kesme şeklinde yapılabilir.

Sorular ve Cevapları

1/6

0:00:04

Seçimi temizle 

Sonraki Soru 

Karar ağaçları için hangisi doğrudur?

Sürekli değerli hedef kavramları öğrenmek için kullanılırlar.

Karar ağaçları bir mantık fonksiyonu şeklinde de ifade edilebilir.

Eğitim verisinin hatalı örnekler içermesi durumunda kullanılamazlar.

Eğitim verisi eksik değerli özellikler içeriyorsa kullanılamazlar.

Sorular ve Cevapları

1/6

0:00:04

Seçimi temizle ↺

Sonraki Soru →

Karar ağaçları için hangisi doğrudur?

Sürekli değerli hedef kavramları öğrenmek için kullanılırlar.

Karar ağaçları bir mantık fonksiyonu şeklinde de ifade edilebilir.

Eğitim verisinin hatalı örnekler içermesi durumunda kullanılamazlar.


Eğitim verisi eksik değerli özellikler içeriyorsa kullanılamazlar.

- a) Karar ağacı öğrenme ayırık (discrete) değerli fonksiyonlara yakınsamak için kullanılan bir metottur.
- b) Genel olarak, karar ağaçlarının ifade ettiği fonksiyonlar mantık operatörleri (ve, veya) ile gösterilebilir. Örneğin, Şekil 4.1'deki karar ağacı, aşağıdaki fonksiyona karşılık gelir:
- c) Eğitim verisi bazı hatalar içerebilir. Karar ağaçları hatalar içeren eğitim verisi ile kullanılabilir. Bu hatalar, hem sınıf değerlerinde hem de özelliklerin değerlerinde olabilir.
- d) Eğitim verisi içinde bazı özelliklerin değerleri olmayabilir. Örneğin bazı günler için elimizde Nem verisi olmayabilir. Bu durumda da karar ağaçları kullanılabilir.

Sorular ve Cevapları

2/6

0:00:19

Seçimi temizle 

← Önceki Soru

Sonraki Soru →

ID3 algoritması için hangisi yanlıştır?

Her düğümde farklı özelliklerin değerlerini kontrol eder.

Oluşturulan düğüm bir daha kontrol edilmez.

Sadece pozitif örnekler dikkate alınır.

Bütün eğitim verisini doğru sınıflandıran bir ağaç döner.

Sorular ve Cevapları

2/6

0:00:19

Seçimi temizle

Önceki Soru

Sonraki Soru

ID3 algoritması için hangisi yanlıştır?

Her düğümde farklı özelliklerin değerlerini kontrol eder.

Oluşturulan düğüm bir daha kontrol edilmez.

Sadece pozitif örnekler dikkate alınır.

Bütün eğitim verisini doğru sınıflandıran bir ağaç döner.

a) ID₃ algoritmasının temeli her düğümde hangi özelliğin test edileceğine dayalıdır.

b) ID₃ çalışırken hiç geriye dönüş (backtrack) yapmaz. Bir özelliğe karar verirse örneğin, bunu daha sonra dönüp tekrar kontrol etmez.

c) Şekil 4.2 ID₃ algoritmasının adımlarında

Bütün örnekler pozitif ise, tek düğümlü ve etiketi + olan Kök ağacını dön

Bütün örnekler negatif ise, tek düğümlü ve etiketi - olan Kök ağacını dön

d) ID₃'nin arama uzayı tamdır (complete). Her ayrık değerli fonksiyon bir karar ağacı ile gösterilebileceği için, ID₃'nin arama uzayı da bütün karar ağaçları uzayı olduğu için, ID₃'nin hedef fonksiyonu bulamama gibi bir problemi yoktur.

Sorular ve Cevapları

3/6

0:00:38

Seçimi temizle

Önceki Soru

Sonraki Soru

[8+, 14-] ile gösterilen eğitim verisinin entropisi kaçtır?

0.5

0.761

0.946

0.825

Sorular ve Cevapları

3/6 0:00:38

Seçimi temizle

Önceki Soru Sonraki Soru

[8+, 14-] ile gösterilen eğitim verisinin entropisi kaçtır?

0.5

0.761

0.946

0.825

- S isimli pozitif ve negatif örnekler içeren bir grup verildiğinde, S'in bu Boole sınıflandırmaya göre entropisi şöyle ifade edilir:
- Burada p_+ S'deki pozitif örneklerin oranı, p_- de negatif örneklerin oranını gösteriyor.

```
>> -(8/22)*log2(8/22)-(14/22)*log2(14/22)
```

```
ans = 0.9457
```

Sorular ve Cevapları

4/6

0:00:53

Seçimi temizle

Önceki Soru

Sonraki Soru

Hangisi ID3 algoritmasının limitasyonlarından biri değildir?

ID3 ayrık değerli ve Boole hedef sınıflı bir kavramı bulamayabilir.

Sadece tek bir hipotezi çıktı olarak verir.

Bazı durumlarda lokal en iyi sonuca ulaşır.

Hedef kavramı sağlayan diğer hipotezleri bulamaz.

Sorular ve Cevapları

4/6

0:00:53

Seçimi temizle

Önceki Soru

Sonraki Soru

Hangisi ID3 algoritmasının limitasyonlarından biri değildir?

ID3 ayrık değerli ve Boole hedef sınıflı bir kavramı bulamayabilir.

Sadece tek bir hipotezi çıktı olarak verir.

Bazı durumlarda lokal en iyi sonuca ulaşır.

Hedef kavramı sağlayan diğer hipotezleri bulamaz.

- Karar ağacı öğrenme, ayrık (discrete) değerli fonksiyonlara yakınsamak için kullanılan bir metottur. Öğrenilen fonksiyon bir karar ağacı olarak ifade edilir. Örnekler, özellik : değer ikilileri şeklinde gösterilirler. Örnekler, bir küme sabit özelliklerle ve onları değerleri ile gösterilirler. Hedef fonksiyonunun değerleri ayrıktır. Şekil 4.1'deki karar ağacı, her örneğe Boole bir değer atıyor. Karar ağaçlarının geliştirilmiş bir versiyonu reel değerlerle de çalışabilir olsa da bu versiyon çok yaygın kullanılmaz.
- Diğer tümevarımsal (inductive) öğrenme metotlarında olduğu gibi, ID3 de bir hipotez uzayındaki en iyi hipotezi arar.

Sorular ve Cevapları

4/6

0:00:53

Seçimi temizle

Önceki Soru

Sonraki Soru

Hangisi ID3 algoritmasının limitasyonlarından biri değildir?

ID3 ayrık değerli ve Boole hedef sınıflı bir kavramı bulamayabilir.

Sadece tek bir hipotezi çıktı olarak verir.

Bazı durumlarda lokal en iyi sonuca ulaşır.

Hedef kavramı sağlayan diğer hipotezleri bulamaz.

- c. ID₃ çalışırken hiç geriye dönüş (backtrack) yapmaz. Bir özelliğe karar verirse örneğin, bunu daha sonra dönüp tekrar kontrol etmez. Bu durumda da bu tip algoritmaların yaşadığı genel sorunla karşılaşabilir : Global olarak değil de lokal olarak bir en iyi sonuca ulaşmak.
- d. ID₃ her an tek bir hipotezi hesaba kattığı için birden fazla hipotez çıktısı veremez.

Sorular ve Cevapları

5/6

0:01:13

Seçimi temizle

Önceki Soru

Sonraki Soru

Aşağıdaki sürekli değerli özellik için hangi eşik değerini seçmek mantıklıdır?

Özellik1	26.2	78.9	54.2	111.2	90.1	10.1
Hedef	Hayır	Hayır	Evet	Evet	Evet	Hayır

54.2

60.65

40.2

50.1

Sorular ve Cevapları

5/6 0:01:13

Seçimi temizle

Önceki Soru Sonraki Soru

Aşağıdaki sürekli değerli özellik için hangi eşik değerini seçmek mantıklıdır?

Özellik1	26.2	78.9	54.2	111.2	90.1	10.1
Hedef	Hayır	Hayır	Evet	Evet	Evet	Hayır

54.2

60.65

40.2

50.1

Özellik1	10.1	26.2	54.2	78.9	90.1	111.2
Hedef	Hayır	Hayır	Evet	Hayır	Evet	Evet

Örnekleri sürekli değerli özellik 1'e göre sıralayıp, daha sonra yan yana olan hangi değerlerde hedef kavramın değiştiğini bulursak, bu değerlerin ortalaması ile bir küme aday eşik değer oluşturabiliriz. Bu aday eşik değerleri kullanarak her biri için bilgi kazanımını hesaplayabiliriz. Yukarıdaki örnekte, bu prosedüre göre, üç aday eşik değer vardır: $(26.2+54.2)/2=40.2$ (ilkini bulduk ve cevap şıklarında var).

Sorular ve Cevapları

5. Soru için gerekli açıklamalar

ID3'de hem hedef kavram, hem de her düğümde test edilen özellikler ayrık değerli olmalıdır. Özellikler üzerindeki kısıt, kolayca kaldırılabilir. Bu sürekli değerlere sahip bir özelliği ayrık değerler kümesine ayırmakla yapılabilir. Yani, herhangi bir sürekli değerli A özelliği için, algoritma dinamik olarak A_c gibi, $A < c$ iken doğru değilken yanlış olan, Boole değerli bir özellik tanımlayabilir. Buradaki tek soru c eşik değerinin nasıl seçileceğidir.


Örnek olarak, Sıcaklık özelliğinin sürekli haline bakalım. Sıcaklık özelliği ve TenisOyna hedef kavramının aşağıdaki gibi değerler aldığını varsayalım.

Sıcaklık	40	48	60	72	80	90
TenisOyna	Hayır	Hayır	Evet	Evet	Evet	Hayır

Burada c'yi bilgi kazanımını maksimum yapacak şekilde seçmemiz gerektiği açıktır. Örnekleri sürekli değerli özellik A'ya göre sıralayıp, daha sonra yan yana olan hangi değerlerde hedef kavramın değiştiğini bulursak, bu değerlerin ortalaması ile bir küme aday eşik değer oluşturabiliriz. Bu aday eşik değerleri kullanarak her biri için bilgi kazanımını hesaplayabiliriz. Yukarıdaki örnekte, bu prosedüre göre, iki aday eşik değer vardır: $(48+60)/2$ ve $(80+90)/2$. Daha sonra bu iki özellik (Sıcaklık₅₄ ve Sıcaklık₈₅) için de bilgi kazanımı hesaplanırsa en iyi özellik bulunacaktır (bu Sıcaklık₅₄tür). Bundan sonra bulunan en iyi özellik aynen diğer özellikler gibi ID3 algoritmasında kullanılabilir.

Sorular ve Cevapları

6/6

Seçimi temizle 



TESTİ BİTİR



Önceki Soru

Bilgi kazanımı ve alternatif özellik seçimi ölçüleri için hangisi yanlıştır?

Bilgi kazanımı, değer kümesi çok olan özellikler için kötü bir ölçü olabilir.

Kazanım oranı bilgi kazanımının bazı eksikliklerini giderebilir.

Ayırma bilgisi aslında bir entropi hesabıdır.

Kazanım oranı, hemen hemen bütün örnekler için aynı değerlere sahip özellikler için de iyi sonuç verir.

Sorular ve Cevapları

6/6

Seçimi temizle



TESTİ BİTİR



Önceki Soru

Bilgi kazanımı ve alternatif özellik seçimi ölçüleri için hangisi yanlıştır?

Bilgi kazanımı, değer kümesi çok olan özellikler için kötü bir ölçü olabilir.

Kazanım oranı bilgi kazanımının bazı eksikliklerini giderebilir.

Ayrırma bilgisi aslında bir entropi hesabıdır.

Kazanım oranı, hemen hemen bütün örnekler için aynı değerlere sahip özellikler için de iyi sonuç verir.

- a. Bilgi kazanımı hesabı, çok değer alabilen özellikleri az değer alabilen özelliklere tercih etmeye meyillidir.
- b. Kazanım oranı, özellikleri, veriyi ne kadar düzenli ayırdıklarına göre de değerlendirir.
- c. Aslında ayırma bilgisi ölçüsünün de A'nın değerlerine göre hesaplanan entropi olduğunu da gözden kaçırmamak gerekir.
- d. Kazanım oranı ile ilgili çıkabilecek bir sorun, ayırma bilgisinin değeri ile ilgilidir. Bu değer bazı durumlarda çok küçük veya 0 olabilir. Bu kazanım oranı değerini ya tanımsız ya da çok büyük bir değer yapar. Bu durum, hemen hemen bütün örnekler için aynı değere sahip özelliklerde ortaya çıkabilir. Böyle bir durumda, önce bütün özellikler için bilgi kazanımı hesaplanıp, ortalamadan yüksek bilgi kazanımı olan özellikler için bir de kazanım oranı hesaplanabilir ve seçim buna göre yapılabilir.

Not

Rapidminer isimli programı kullanarak karar ağacı algoritmalarını uygulayabilirsiniz