Makine Öğrenmesi

SD413

Karar Ağaçları

- Önceki bölümlerde tartışılan sınıflandırıcılar, tüm öznitelik değerlerinin aynı anda bilinmesini gerektirir. Bu sınıflandırıcılar, örneği açıklayan eksiksiz öznitelik vektörüne dayalıdır.

- Bazı uygulamalarda bu senaryo gerçekçi değildir. Hastasının rahatsızlığının nedenini arayan bir hekimin, birkaç sübjektif semptom dışında başlayacak hiçbir şeyi olmayabilir.

- Muhtemel teşhis alanını daraltmak için birkaç laboratuvar testi ve bunların sonuçlarına göre ek laboratuvar testleri gerekebilir. Başka bir deyişle, doktor yalnızca anlık olarak katkıda bulunabilecek "nitelikleri" dikkate alır ve geri kalan özellikleri göz ardı eder.

Karar Ağaçları

- Öznitelik değerleri hakkında ayrıntılı bilgi hemen elde edilemeyebilir ve hatta sınıflandırıcının her seferinde bir özniteliğe odaklanması ve her zaman maksimum alakalı bilgi sunan özniteliği seçmesi koşuluyla, buna ihtiyaç bile olmayabilir.

- Bu düşünceler etrafında oluşturulmuş popüler bir araç, karar ağacı olarak bilinir.

Karar Ağaçları

- Tablodaki eğitim seti, üç özellik tarafından açıklanan ve belirli bir sınıfın olumlu veya olumsuz örnekleri olarak etiketlenen sekiz örnekten oluşur. Kolaylık sağlamak için, şimdilik tüm niteliklerin ayrık (discrete) olduğunu varsayalım.

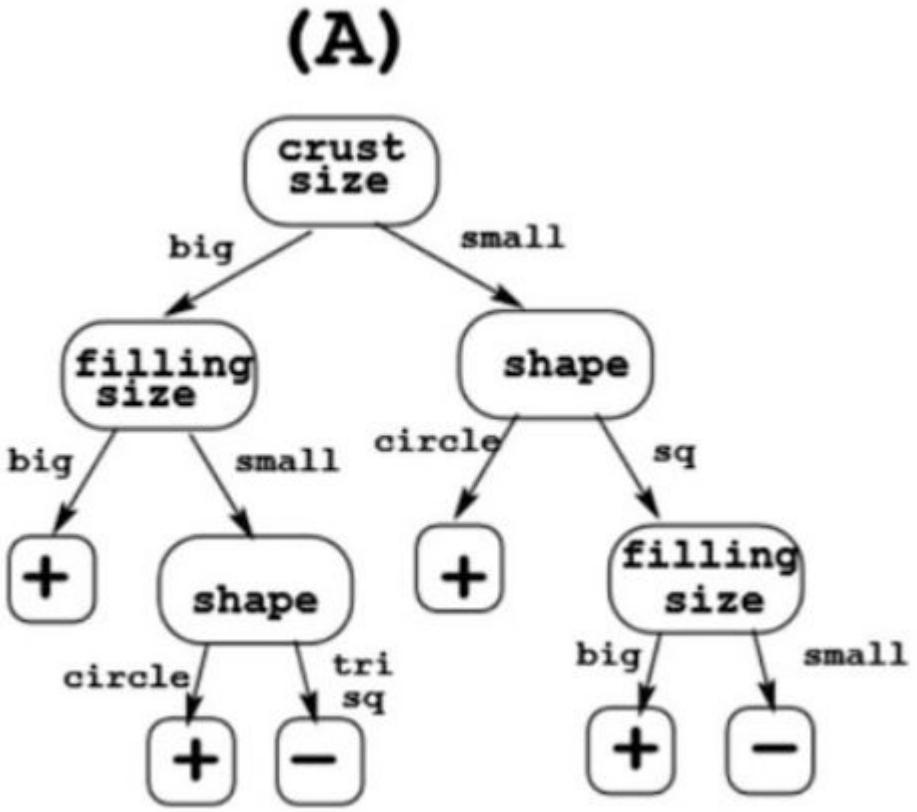
Example	crust size	shape	filling size	Class
*e*1	big	circle	small	**pos**
*e*2	small	circle	small	**pos**
*e*3	big	square	small	**neg**
*e*4	big	triangle	small	**neg**
*e*5	big	square	big	**pos**
*e*6	small	square	small	**neg**
*e*7	small	square	big	**pos**
*e*8	big	circle	big	**pos**

Karar Ağaçları

- Sonraki slaytlarda, tablodaki verileri doğru şekilde sınıflandıran birkaç örnek karar ağacı gösterilmektedir.
- Düğümler nitelik-değer testleridir, kenarlar test sonuçlarına göre nasıl ilerleneceğini gösterir ve yapraklar sınıf etiketlerini temsil eder.
- Sınıflandırılacak bir örnek önce en üst düğüm olan kökte belirtilen teste tabi tutulur. Testin sonucu daha sonra kökten hangi kenarın takip edileceğine karar verir.
- İşlem, bir yaprak düğüme ulaşılana kadar devam eder ve ardından örnek, bu yaprakla ilişkili sınıfla etiketlenir.

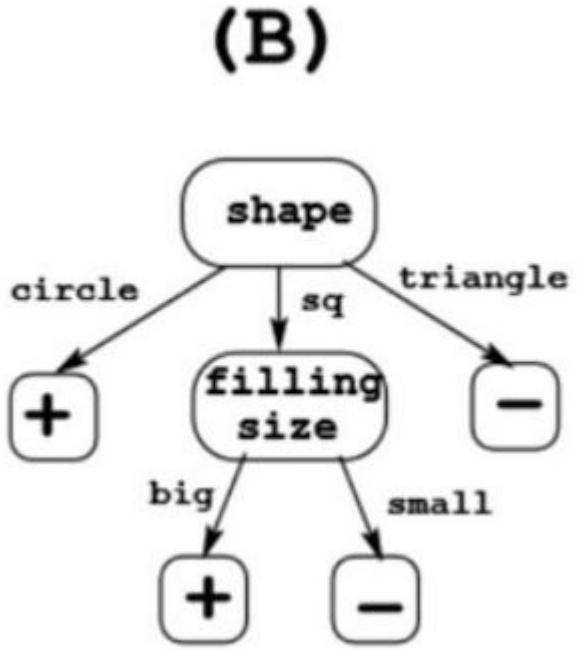
Karar Ağaçları

Example	crust size	shape	filling size	Class
*e*1	big	circle	small	**pos**
*e*2	small	circle	small	**pos**
*e*3	big	square	small	**neg**
*e*4	big	triangle	small	**neg**
*e*5	big	square	big	**pos**
*e*6	small	square	small	**neg**
*e*7	small	square	big	**pos**
*e*8	big	circle	big	**pos**



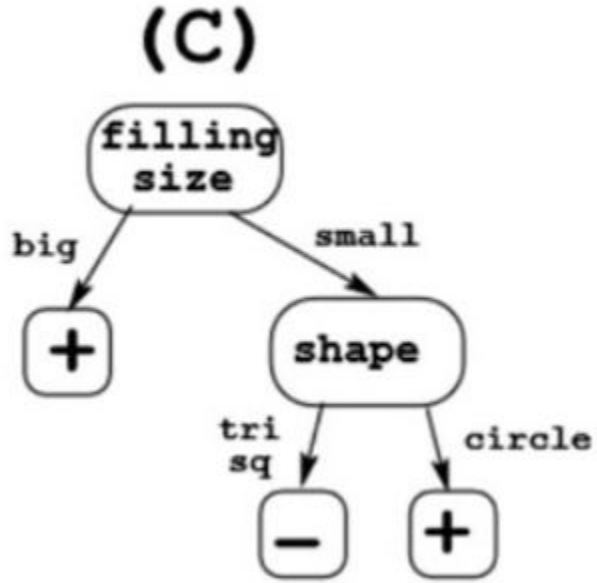
Karar Ağaçları

Example	crust size	shape	filling size	Class
*e*1	big	circle	small	**pos**
*e*2	small	circle	small	**pos**
*e*3	big	square	small	**neg**
*e*4	big	triangle	small	**neg**
*e*5	big	square	big	**pos**
*e*6	small	square	small	**neg**
*e*7	small	square	big	**pos**
*e*8	big	circle	big	**pos**



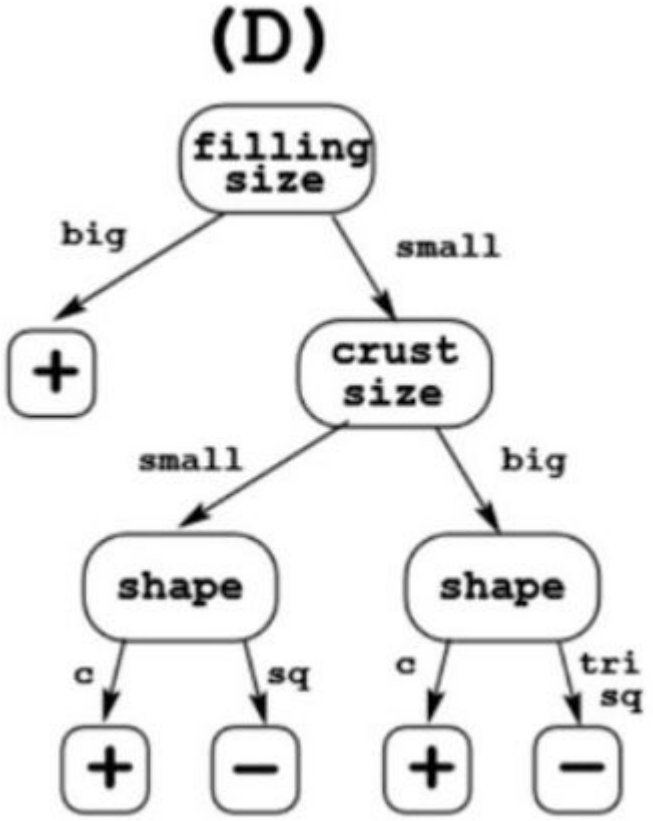
Karar Ağaçları

Example	crust size	shape	filling size	Class
*e*1	big	circle	small	**pos**
*e*2	small	circle	small	**pos**
*e*3	big	square	small	**neg**
*e*4	big	triangle	small	**neg**
*e*5	big	square	big	**pos**
*e*6	small	square	small	**neg**
*e*7	small	square	big	**pos**
*e*8	big	circle	big	**pos**



Karar Ağaçları

Example	crust size	shape	filling size	Class
*e*1	big	circle	small	**pos**
*e*2	small	circle	small	**pos**
*e*3	big	square	small	**neg**
*e*4	big	triangle	small	**neg**
*e*5	big	square	big	**pos**
*e*6	small	square	small	**neg**
*e*7	small	square	big	**pos**
*e*8	big	circle	big	**pos**



Karar Ağaçları

- Bu sınıflandırıcıyı önceki bölümlerde tartışılanlarla karşılaştırdığımızda, bir avantaj görüyoruz: yorumlanabilirlik. Biri e1 örneğinin neden pozitif olduğunu sorarsa, cevap "çünkü şekli dairedir" olur. Diğer paradigmalar açıklamalar sunmaz.

- Bayesci ve lineer sınıflandırıcılar tipik kara kutulardır: bir örnekle sunulduklarında, basitçe sınıfını döndürürler ve asla herhangi bir neden sunmazlar. İlkel bir argüman görünümü sunan k-NN sınıflandırıcısı durumunda durum yalnızca biraz daha iyidir: örneğin, "x pozitiftir çünkü bu, x'e en çok benzeyen eğitim örneğinin sınıfıdır."

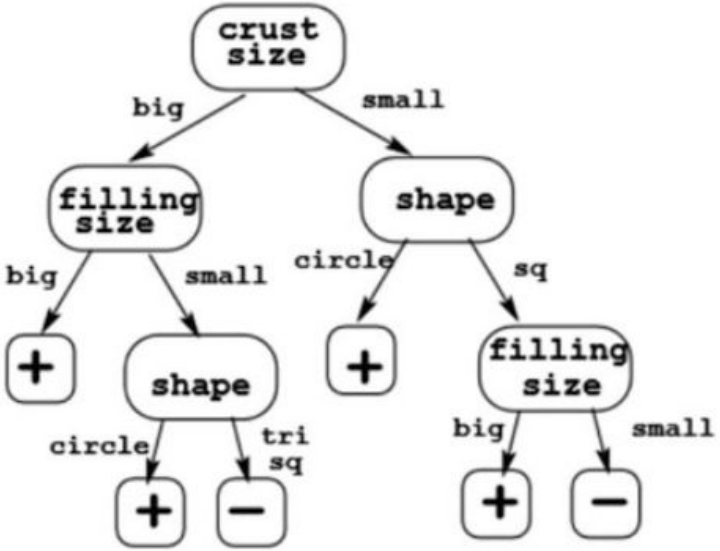
Karar Ağaçları

- Bir karar ağacını “şekil=kare ve dolgu-boyutu=büyükse, o zaman pozitif sınıfını seç” gibi bir kurallar dizisi olarak yorumlayabiliriz. Bu kuralları inceleyen bir alan uzmanı bize bunların mantıklı olup olmadığını ve verilen alanla ilgili kendi görüşüne katılıp katılmadıklarını söyleyebilir.

- Örneğin, haşhaş kullanıldığında dolgu siyah, peynir kullanıldığında beyaz olabilir ve somut bir kuralın incelenmesi Johnny'nin haşhaşı peynire tercih ettiği sonucuna götürebilir. Bu anlamda, karar ağaçlarının kullanımı yararlı yeni bilgiler üretebilir.

Karar Ağaçları

- Karar ağaçlarında karşılaşılabilecek problemlerden biri yalnızca veride gözlenen durumların yansıtılmak zorunda oluşudur. Bu aslında diğer sınıflandırıcılar için de bir problemdir. Bir düğümde test edilen niteliğin bütün olası değerleri için düğümden aşağıya doğru kenarlarla gösterilmeyebilir. Bunun nedeni ilgili karar patikasında söz konusu durumun veride gözlenmemesidir.
- Örneğin a- karar ağacında crust-size=small olan örneklerin içinde shape=triangle bulunmadığı için ilgili kenar bulunmaz. Peki ya test ederken rastlanılırsa?

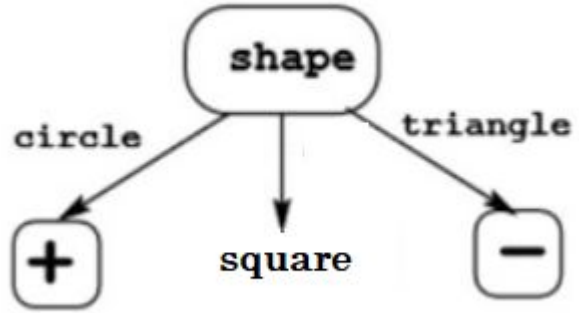


Karar Ağacını İnşa Etmek

Böl ve Fethet Yöntemi:

- Rastgele bir öznitelik seçerek başlayalım: Shape
- Üç değeri var: circle, square, triangle
- Circle ve triangle için diğer nitelikleri test etmeye gerek yok. Ancak square için yeni bir düğüm daha oluşturmak gerekiyor.

Example	crust size	shape	filling size	Class
*e*1	big	circle	small	**pos**
*e*2	small	circle	small	**pos**
*e*3	big	square	small	**neg**
*e*4	big	triangle	small	**neg**
*e*5	big	square	big	**pos**
*e*6	small	square	small	**neg**
*e*7	small	square	big	**pos**
*e*8	big	circle	big	**pos**



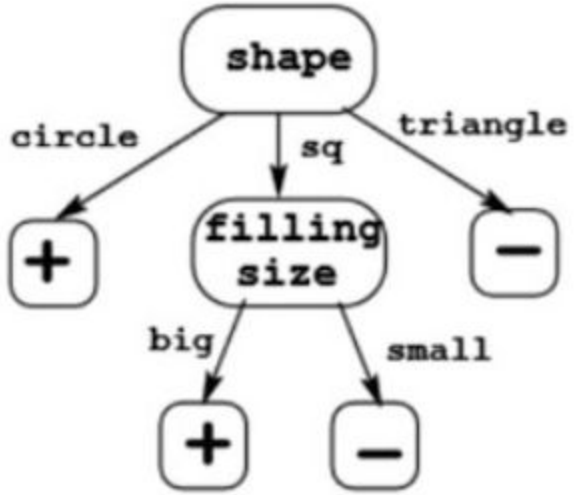
Karar Ağacını İnşa Etmek

Böl ve Fethet Yöntemi:

- İkinci öznitelik olarak filling-size'ı seçelim.

- shape=square iken
filling-size=big olduğunda sonuç +
filling-size=small olduğunda sonuç – olacaktır.

Example	crust size	shape	filling size	Class
*e*1	big	circle	small	**pos**
*e*2	small	circle	small	**pos**
*e*3	big	square	small	**neg**
*e*4	big	triangle	small	**neg**
*e*5	big	square	big	**pos**
*e*6	small	square	small	**neg**
*e*7	small	square	big	**pos**
*e*8	big	circle	big	**pos**



Karar Ağacını İnşa Etmek

- Aynı veriyi çok sayıda farklı karar ağacıyla modellemek mümkündür.

- Bu ağaçlardan hangisi seçilmelidir?
  - **Kriterler**: düğüm sayısı, test sayısı.

Karar Ağacını İnşa Etmek

- Küçük karar ağaçlarının (birkaç testi olanlar) tercih edilmesinin birkaç nedeni vardır. Bunlardan biri yorumlanabilirliktir. İnsan uzmanlar, birkaç testten fazla olmayan bir karar ağacını analiz etmeyi, açıklamayı ve hatta belki de düzeltmeyi kolay bulur. Ağaç ne kadar büyükse, bu o kadar zor olur.

- Küçük karar ağaçlarının bir başka avantajı da ilgisiz ve gereksiz bilgileri ortadan kaldırma eğilimleridir. a'daki nispeten büyük ağaç üç özelliğin tümünü kullanırken, b'deki daha küçük ağaç, kabuk boyutunu hiç dikkate almadan eğitim setini sınıflandırmada aynı derecede iyidir. Bu tür bir ekonomi, belirli öznitelik değerlerinin elde edilmesinin pahalı veya zaman alıcı olduğu alanlarda kullanışlı olacaktır.

- Son olarak, daha büyük ağaçlar eğitim setine gereğinden fazla uyma (overfitting) eğilimindedir.

Karar Ağacını İnşa Etmek

- Böl ve fethet tekniğinin davranışını manuel ağaç oluşturma ile gösterirken, nitelikleri rastgele seçtik ve ardından bazı seçimlerin daha küçük ağaçlarla sonuçlandığını gözlemledik.

- Görünüşe göre, nitelikler aktardıkları bilgi miktarında farklılık gösteriyor. Örneğin, shape bazı örnekleri hemen pozitif (değer circle ise) veya negatif (değer triangle ise) olarak etiketleyebilir; ancak crust-size, başka bir özellik tarafından desteklenmedikçe bunu yapamaz.

- Her özelliğin sunduğu bilgi miktarını ölçmenin bir yolu olduğunu varsayarak bir sonraki slaytta verilen sözde kodla karar ağaçları için bir tümevarım tekniği sunabiliriz.

Karar Ağacını İnşa Etmek

Eğitim seti T olsun.
çoğalt(T):

(1) Sınıf etiketleri hakkında maksimum bilgiye katkıda bulunan özniteliği bul.
(2) T'yi, her biri farklı bir değerle karakterize edilen T_i alt kümelerine ayır.
(3) Her T_i için:
Eğer T_i'deki tüm örnekler aynı sınıfa aitse, bu sınıfla etiketlenmiş bir yaprak oluştur;
aksi takdirde, aynı prosedürü her bir eğitim alt kümesine tekrar tekrar uygula: Grow(T_i).

Karar Ağacını İnşa Etmek

Bir Nitelikte Ne Kadar Bilgi Vardır?

- Eğitim örneklerinin pos veya neg olarak etiketlendiğini ve eğitim setindeki bu iki sınıfın olasılıklarının ppos ve pneg olduğunu varsayalım. Rastgele bir eğitim örneği seçelim. "Bu örneğin sınıfı pos" mesajı ile ne kadar bilgi aktarılıyor?

- Cevap ppos'a bağlıdır. Tüm örneklerin pozitif olduğunun bilindiği uç durumda, ppos = 1, mesaj bize yeni bir şey söylemez. Öyle söylenmese de örneğin olumlu olduğunu biliyoruz; bu mesajdaki bilgi miktarı sıfırdır.

- Her iki sınıf eşit olarak temsil edildiğinde durum farklıdır, ppos = pneg = 0.5. Burada tahminimiz yazı tura atmaktan daha iyi değil ve sınıf etiketiyle ilgili bir mesaj bazı bilgiler sağlıyor.

Karar Ağacını İnşa Etmek

Bir Nitelikte Ne Kadar Bilgi Vardır?

- Ve örneklerin büyük çoğunluğunun negatif olduğu biliniyorsa, örneğin ppos = 0.01 ise, seçilen örneğin negatif olduğundan neredeyse eminiz. Durumun böyle olmadığı mesajı beklenmedik, yani bilgi içeriği yüksek. ppos değeri ne kadar düşükse mesajın o kadar çok bilgi taşıdığını görüyoruz. Bu tür bir mesajın bilgi içeriğini ölçerken, aşağıdaki formül kullanılır:

$$I_{\text{pos}} = -\log_2 p_{\text{pos}}$$

p_{pos}	$-\log_2 p_{\text{pos}}$
1.00	0 bits
0.50	1 bit
0.25	2 bits
0.125	3 bits

- Negatif işaret, ppos $\in$ (0, 1)'in logaritmasının her zaman negatif olması durumunu telafi eder.

- Bilgi miktarının birimi 1 bit'tir. Başka bir yorum: logaritmanın tabanı her zaman 2'dir, log2 ppos yerine log ppos yazmak yaygındır.

Entropi: Ortalama Bilgi İçeriği

- Tek bir mesajın bilgi içeriği bu şekilde hesaplanır. Şimdi, birinin her seferinde bir eğitim örneği seçtiğini ve tüm eğitim örnekleri kontrol edilene kadar bize her zaman sınıf etiketinin ne olduğunu söylediğini varsayalım.

- İlki ppos olasılıkla, ikincisi pneg olasılıkla "örnek pozitif" ve "örnek negatif" olmak üzere her iki mesaj da oluşacaktır. Ortalama bilgi içeriklerini hesaplamak için, iki mesajın her birinin bilgi içeriklerini eğitim seti T'deki olasılıklarına (veya göreli frekanslarına) göre tartarız:

$$H(T) = -p_{\text{pos}} \log_2 p_{\text{pos}} - p_{\text{neg}} \log_2 p_{\text{neg}}$$

Karar Ağacını İnşa Etmek

- Burada bir sorun var gibi görünüyor: sıfırın logaritması tanımlı değil ve entropi denklemi ppos = 0 veya pneg = 0 ise anlamsızdır.

- Neyse ki, basit bir analiz (limitleri ve l'Hopital kuralını kullanarak), p'nin sıfıra yakınsaması için p log p ifadesinin de sıfıra yakınsadığını ortaya çıkaracaktır ve 0 · log 0 = 0 olduğu sonucuna varırız.

Karar Ağacını İnşa Etmek

- H(T), T'nin entropisi olarak adlandırılır. Değeri, ppos = pneg = 0.5 olduğunda maksimum değerine (çünkü 0.5 · log 0.5 + 0.5 · log 0.5 = 1) , H(T) = 1'e ulaşır; ve ppos = 1 veya pneg = 1 olduğunda minimum değeri olan H(T) = 0'a düşer (çünkü 0 · log 0 + 1 · log 1 = 0).

- H(T) = 0 ile sonuçlanan ppos = 1 veya pneg = 1 durumu, kümedeki tüm örnekler aynı sınıfa ait olduğu için mükemmel düzenlilik olarak kabul edilir. Tersine, H(T) = 1 ile sonuçlanan ppos = pneg = 0.5 durumu, herhangi bir düzenliliğin tamamen yokluğu olarak görülür. Bu nedenle entropinin bazen verilerdeki "kaos" miktarını ölçtüğü söylenir.

Karar Ağacını İnşa Etmek

Bir Özniteliğin Katkıda Bulunduğu Bilgi Miktarı

- Entropi kavramı (düzensizlik) önemli bir soruyla yüzleşmemize yardımcı olacaktır: Ayrık bir özniteliğin değerinin bilgisi bize bir örneğin sınıfı hakkında ne kadar bilgi verir?

- a özniteliğinin, eğitim kümesini (T), her biri farklı bir a değeriyle karakterize edilen alt kümelere (T_i) böldüğünü hatırlayalım.

- Doğal olarak, her bir alt küme, iki sınıfın, ppos ve pneg'in kendi olasılıkları (göreli frekansları) tarafından işaretlenecektir. Bu değerleri entropi formülüne uygularsak, her bir alt kümenin entropisini, $H(T_i)$ elde ederiz.

Karar Ağacını İnşa Etmek

Bir Özniteliğin Katkıda Bulunduğu Bilgi Miktarı

- $|T_i|$ T_i'deki örneklerin sayısı olsun ve $|T|$ tüm eğitim setindeki (T) örneklerin sayısı olsun. Rastgele çekilmiş bir eğitim örneğinin T_i'ye ait olma olasılığı şu şekilde tahmin edilir:

$$P_i = \frac{|T_i|}{|T|}$$

- Bununla, tüm entropilerin ağırlıklı ortalamasını hesaplayabiliriz:

$$H(T, a) = \Sigma_i P_i \cdot H(T_i)$$

Bir Özniteliğin Katkıda Bulunduğu Bilgi Miktarı

- Sonuç, H(T, a), her bir eğitim örneği için yalnızca sınıf etiketinin değil, a niteliğinin değerinin de bilindiği bir sistemin entropisidir. a tarafından sağlanan bilgi miktarı, a dikkate alınmadan ölçülen sistemin entropisi ile bu öznitelik dikkate alındığında ortaya çıkan sistemin entropisi arasındaki farktır:

$$I(T, a) = H(T) - H(T, a)$$

Bir Özniteliğin Katkıda Bulunduğu Bilgi Miktarı

- Bu farkın negatif olamayacağını kanıtlamak kolaydır; bilgi ancak dikkate alınarak kazanılabilir, asla kaybolmaz. Bununla birlikte, bazı nadir durumlarda, I (T, a) = 0, yani a, eğitim örneklerinin sınıf etiketleri hakkında herhangi bir bilgi vermez.

- Eşitliği her özniteliğe ayrı ayrı uygulayarak, hangisinin maksimum miktarda bilgi sağladığını ve dolayısıyla algoritmanın ilk adımındaki "kök" testi için en iyi seçim olduğunu belirleyebiliriz. En iyi nitelik seçme prosedürü sonraki slaytta verilen sözde kodla özetlenmiştir. Süreç, sadece sınıf yüzdelerinin bilindiği sistemin entropisinin hesaplanması ile başlar. Daha sonra, algoritma her öznitelik tarafından sunulan bilgi kazancını hesaplar. En yüksek bilgi kazancına yol açan özellik en iyisidir.

Karar Ağacını İnşa Etmek

En bilgilendirici özelliği bulmak için algoritma

1. Pozitif ve negatif örneklerin yüzdelerini, ppos ve pneg'i kullanarak, eğitim kümesinin entropisini (T) hesaplayın:

$$H(T) = -ppos \log_2 ppos - pneg \log_2 pneg$$

2. T'yi P_i olasılığıyla T_i altkümelerine bölen her a özniteliği için aşağıdakileri yapın:

   i. her T_i alt kümesi için entropiyi hesaplayın ve $H(T_i)$ ile gösterin;
   ii. sistemin ortalama entropisini hesaplayın: $H(T, a) = \sum_i P_i \cdot H(T_i)$;
   iii. bilgi kazancını hesaplayın: $I(T, a) = H(T) - H(T, a)$

3. En yüksek bilgi kazancına sahip özelliği seçin.

Karar Ağacını İnşa Etmek

Sayısal Örnek

- Başlangıçta, öznitelikleri olmayan sistemin entropisi H(T) belirlenir. Daha sonra, shape'in eğitim setini üç alt kümeye böldüğünü gözlemliyoruz.
- Entropilerinin ortalaması, H(T, shape), hesaplanır ve H(T) ile H(T, shape) arasındaki fark, bu özelliğin aktardığı bilgiyi verir.
- Crust-size ve Filling-size için prosedür tekrarlanır.

Sayısal Örnek

Example	crust size	shape	filling size	Class
*e*1	big	circle	small	**pos**
*e*2	small	circle	small	**pos**
*e*3	big	square	small	**neg**
*e*4	big	triangle	small	**neg**
*e*5	big	square	big	**pos**
*e*6	small	square	small	**neg**
*e*7	small	square	big	**pos**
*e*8	big	circle	big	**pos**

- Yalnızca sınıf etiketlerinin bilindiği eğitim setinin entropisi:

$$H(T) = -p_{pos} \log_2 p_{pos} - p_{neg} \log_2 p_{neg}$$
$$= -(5/8) \log(5/8) - (3/8) \log(3/8) = 0.954$$

- Daha sonra, shape değerleri tarafından tanımlanan alt kümelerin entropilerini hesaplıyoruz:

$$H(\text{shape} = \text{square}) = -(2/4) \cdot \log(2/4) - (2/4) \cdot \log(2/4) = 1$$
$$H(\text{shape} = \text{circle}) = -(3/3) \cdot \log(3/3) - (0/3) \cdot \log(0/3) = 0$$
$$H(\text{shape} = \text{triangle}) = -(0/1) \cdot \log(0/1) - (1/1) \cdot \log(1/1) = 0$$

- Şimdi bilinen sınıf etiketleri ve shape olan bir sistemin ortalama entropisini elde ediyoruz:

$$H(T, \text{shape}) = (4/8) \cdot 1 + (3/8) \cdot 0 + (1/8) \cdot 0 = 0.5$$

Sayısal Örnek

Example	crust size	shape	filling size	Class
*e*1	big	circle	small	**pos**
*e*2	small	circle	small	**pos**
*e*3	big	square	small	**neg**
*e*4	big	triangle	small	**neg**
*e*5	big	square	big	**pos**
*e*6	small	square	small	**neg**
*e*7	small	square	big	**pos**
*e*8	big	circle	big	**pos**

- Şimdi bilinen sınıf etiketleri ve shape olan bir sistemin ortalama entropisini elde ediyoruz:

  $H(T, \text{shape}) = (4/8) \cdot 1 + (3/8) \cdot 0 + (1/8) \cdot 0 = 0.5$

- Diğer iki özellik için aynı prosedürü tekrarlayarak aşağıdakileri elde ederiz:

  $H(T, \text{crustsize}) = 0.951$
  $H(T, \text{crustsize}) = 0.607$

- Aşağıdaki değerler, tüm nitelikler için bilgi kazanımlarını verir:

  $I(T, \text{shape}) = H(T) - H(T, \text{shape}) = 0.954 - 0.5 = 0.454$
  $I(T, \text{crustsize}) = H(T) - H(T, \text{crustsize}) = 0.954 - 0.951 = 0.003$
  $I(T, \text{fillingsize}) = H(T) - H(T, \text{fillingsize}) = 0.954 - 0.607 = 0.347$

Sayısal Örnek

Example	crust size	shape	filling size	Class
*e*1	big	circle	small	**pos**
*e*2	small	circle	small	**pos**
*e*3	big	square	small	**neg**
*e*4	big	triangle	small	**neg**
*e*5	big	square	big	**pos**
*e*6	small	square	small	**neg**
*e*7	small	square	big	**pos**
*e*8	big	circle	big	**pos**

I(T, shape) = H(T) – H(T, shape) = 0.954 – 0.5 = 0.454
I(T, crustsize) = H(T) – H(T, crustsize) = 0.954 – 0.951 = 0.003
I(T, fillingsize) = H(T) – H(T, fillingsize) = 0.954 – 0.607 = 0.347

- Sonuçları karşılaştırdığımızda, shape özniteliğinin daha fazla bilgiye katkıda bulunduğunu fark ederiz.
- Böylece kök testi için shape'i seçiyoruz.

Karar Ağacını İnşa Etmek

- Karar ağaçlarının avantajlarından biri de davranışlarının kolayca yorumlanabilmesidir.
- Kökten yaprağa doğru devam eden test dizisi, eğer-ise kuralını temsil eder.
- Bu Kural sınıflandırıcının neden belirli bir sınıf etiketini seçtiğini açıklar.

if crust-size=big AND filling-size=big then pos
if crust-size=big AND filling-size=small AND shape=circle then pos

Neden Karar Ağaçları?

- Karar ağaçları, kanıtlanmış sınıflandırma performansı sayesinde popülerlik kazanmış diğer yaklaşımların gölgesinde kaldığından, günümüzde eskisinden daha az popülerdir.

- Ancak sağduyulu bir mühendis, modanın dayatmalarına direnir ve bunun yerine bu paradigmanın sunduğu somut faydalara odaklanır. İşte dikkate alınması gereken bazı fikirler:

Neden Karar Ağaçları?

Nitelik Değerlerinin Kullanılabilirliği

- Her şeyden önce, karar ağacı her seferinde bir özellik değeri ister. Bu, değerlerin hemen elde edilemediği ve elde edilmesinin zor ve/veya pahalı olduğu uygulamalarda büyük bir avantajdır.

- Bu nedenle tıbbi teşhiste laboratuvar testleri sadece ihtiyaç duyulduğu kadar istenir; baştan itibaren tüm bu tür testleri içeren kapsamlı bir öznitelik vektörü beklemek saçma olurdu. Bu tür alanlar, k-NN veya lineer sınıflandırıcılar gibi yaklaşımları fiilen diskalifiye eder.

Neden Karar Ağaçları?

Nitelik Seçimi

- En bilgilendirici özelliği seçen mekanizmanın uygulama olanakları yelpazesi karar ağaçları paradigmasının çok ötesine geçer.

- Bu görevi ele alan filtre tabanlı yaklaşım, genellikle karar ağaçları bağlamında kullanılan enformasyon formüllerine dayanır.

Neden Karar Ağaçları?

Üst Düzey Özellikler Oluşturma

- Daha önce sınıflandırma için gerekli bilgiden yoksun olan düşük seviyeli niteliklere sahip vektörlerin sınırlamalarına işaret etmek için kanguru kavramı örneğini kullanmıştık.

- Sonuç olarak, makine öğrenmesinde, mevcut özniteliklerden türetilen daha üst düzey özellikler oluşturmak için bazı mekanizmalara ihtiyaç duyulduğunu belirtmiştik.

- Gelecek bölümlerin bazılarında göreceğimiz gibi, çağdaş makine öğrenmesinin gücü genellikle bu üst düzey özellikleri yaratmaya yönelik üstü kapalı yeteneklerle açıklanır. Olasılıklardan biri, uyarılmış karar ağacının her bir dalını, nitelik-değer testlerinin birleşimiyle tanımlanan yeni bir özellik olarak yorumlamaktır.

Neden Karar Ağaçları?

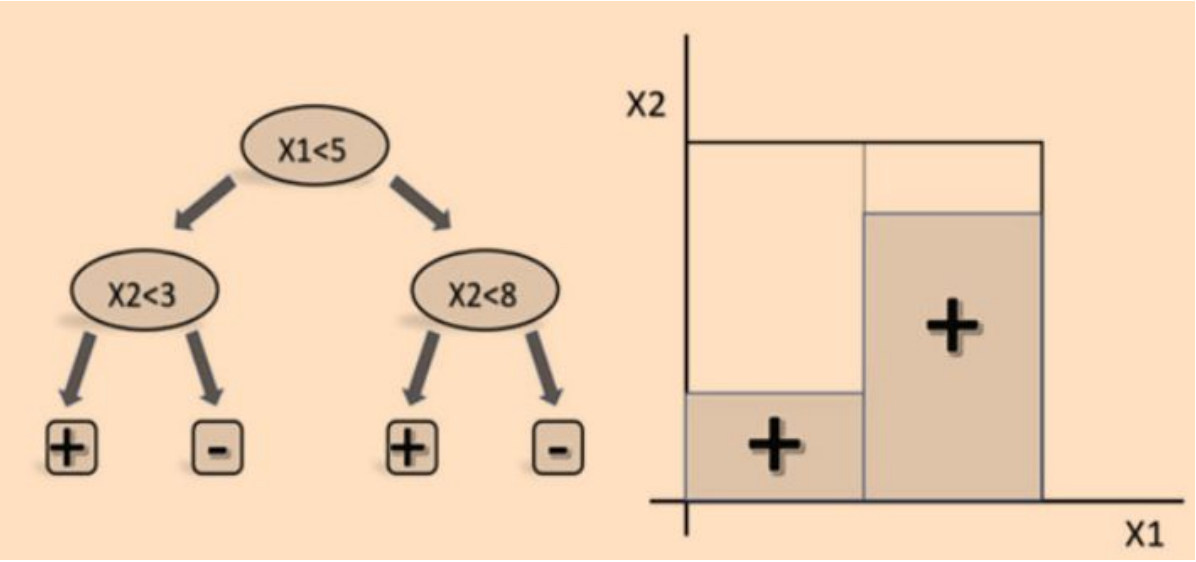
Yapay Sinir Ağlarında Enformasyon Tabanlı Kayıp

- Bir örneğin sınıfıyla ilgili bir mesajda yer alan bilgiyi ölçmek için kullanılan formül bazen yapay sinir ağlarını eğitmek için algoritmalar tarafından kullanılır.

Neden Karar Ağaçları?

Karar Ağacı Bir Karar Yüzeyini Tanımlar

- Doğrusal ve polinom sınıflandırıcıların tanımlayıcı yönü, pozitif sınıftaki örnekleri negatif sınıfa ait olanlardan ayıran bir karar yüzeyini harekete geçirme yetenekleriydi. Aşağıdaki şekil, bizi karar ağacının da farklı nitelikteki karar yüzeylerini tanımladığına ikna edecektir.



Sorular

- Diğer birçok makine öğrenmesi paradigmasına kıyasla karar ağaçlarının temel faydası nedir?
- Karar ağaçlarının tümevarım bilgisi, üst düzey özellik yaratmanın öznitelik seçimi gibi görevlerde bize nasıl yardımcı olabilir?
- Doğrusal bir sınıflandırıcı tarafından tanımlanan karar yüzeyi ile bir karar ağacı tarafından tanımlanan karar yüzeyi arasındaki farkı açıklayın.

- Ders Sonu.