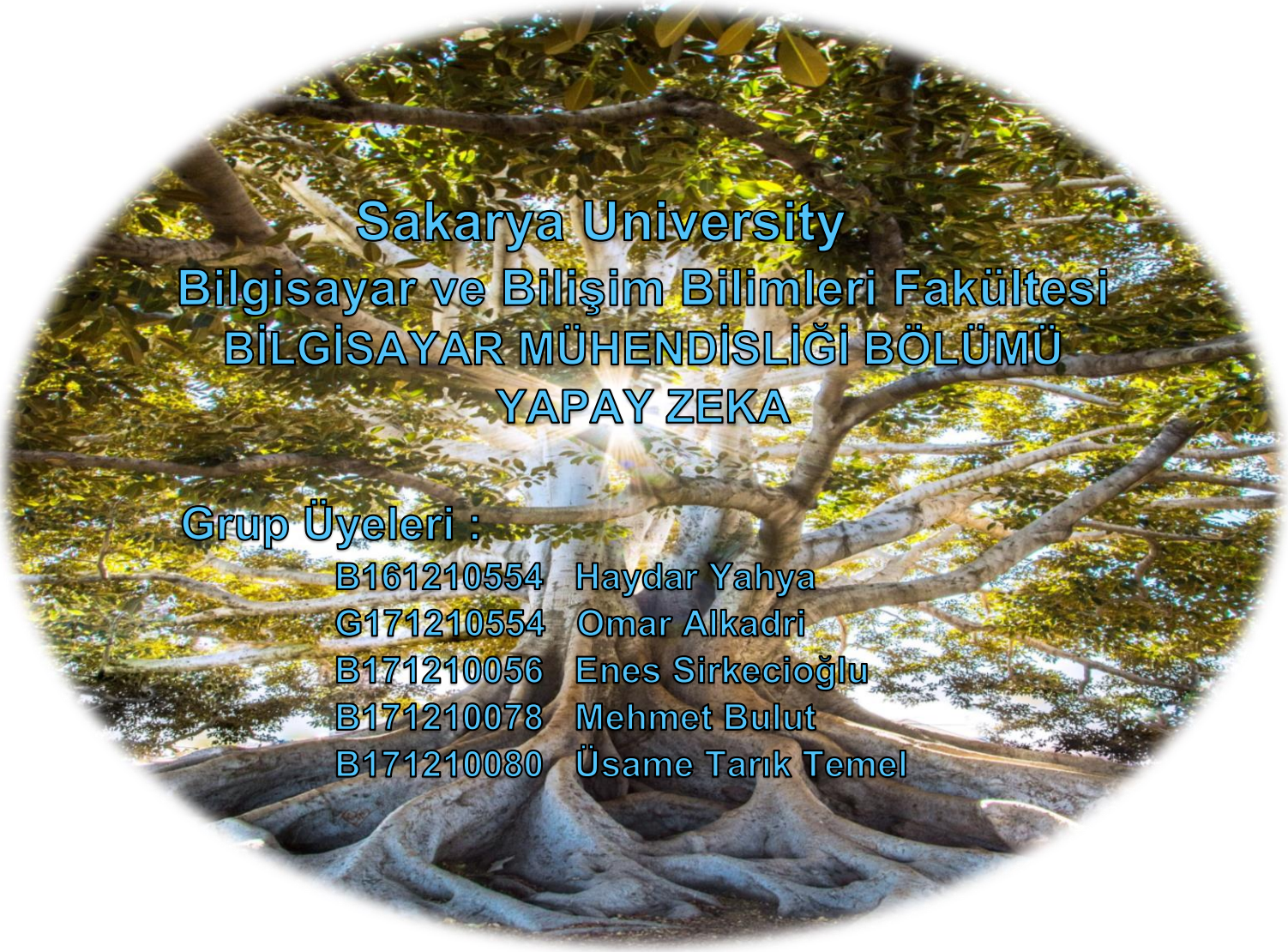
Decision trees (Karar ağaçı)

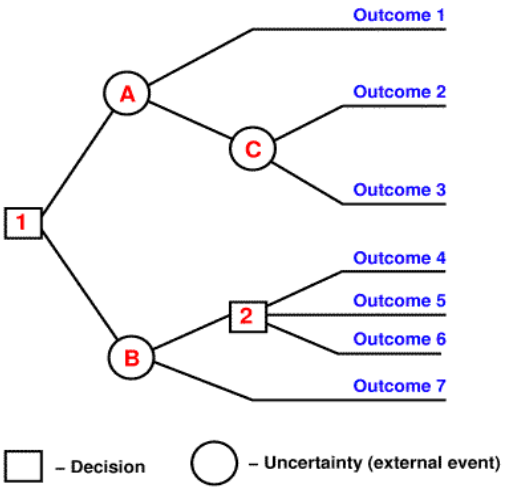


Sydney Üniversitesi'ndeki doktora adayı, Occam'ın Razor adlı önerdiği bir fikir üzerinde duruyordu: "Varlıklar gereksiz yere çoğaltılmamalıdır." Adı J. Ross Quinlan'dı.

Quinlan 1975 yılında Machine Learning'de Occam'ın Razor temelli bir algoritmasını sundu. Buna Yinelemeli Dichotomiser 3 (ID3) adını verdi. Bu algoritma, günümüzde Karar Ağacı adıyla anılan en ünlü algoritmalardan biri olmaya devam edecekti.

Decision trees (Karar ağaçı)

Karar ağaçlarının kökenleri, yazılı kayıtların en erken gelişim çağına kadar gider. Bu tarih, ağaçların büyük bir gücünü gösterir: sezgisel bir yapıya sahip olan ağaç benzeri ve sonuçların anlaşılmasını artıran son derece yorumlanabilir sonuçlar. Karar ağaçlarının (bazen sınıflandırma ağaçları veya regresyon ağaçları olarak da adlandırılır) hesaplama kökenleri biyolojik ve bilişsel süreçlerin modelleridir. Bu ortak miras, hem istatistiksel karar ağaçlarının hem de makine öğrenimi için tasarlanmış ağaçların tamamlayıcı gelişmelerini yönlendirir.



20. yüzyılın sonlarında erken tarih boyunca ağaçların çeşitli özelliklerinin ortaya çıkışı ve aşamalı olarak açıklanması, ilgili önemli referans noktaları ve sorumlu yazarlarla birlikte tartışılmaktadır. Hipotez testi ve çeşitli yeniden örnekleme yaklaşımları gibi istatistiksel yaklaşımlar, makine öğrenme uygulamaları ile birlikte ortaya çıkmıştır. Bu, çeşitli düzeylerde ölçüm ve çeşitli veri kalitesi ile çeşitli istatistiksel ve makine öğrenimi görevleri için uygun olan, son derece uyarlanabilir karar ağacı araçlarına yol açmıştır. Ağaçlar eksik verilerin varlığında sağlamdır ve sonuçtaki modellere eksik verileri dahil etmenin birden fazla yolunu sunar. Ağaçlar güçlü olmasına rağmen, aynı zamanda esnek ve kullanımı kolay yöntemlerdir. Bu da, az sayıda varsayım gerektiren yüksek kaliteli sonuçların üretilmesini sağlar.

Anahtar Kelimeler: karar ağaçları; kural indüksiyonu; öngörücü modeller; makine öğrenmesi;

GİRİŞ

Karar ağaçları, elektronik devrelerin benimsenmesi sırasında elektronik biçimde uygulanacak ilk istatistiksel algoritmalar arasında yer alan genel amaçlı tahmin ve sınıflandırma mekanizmalarıdır.

Karar ağaçlarının ana özelliği, bölüm oluşturmak için ilişkili giriş alanlarının veya öngörücülerin ve ilişkili azalan veri alt kümelerinin (yapraklar veya düğümler olarak adlandırılır) değerlerine göre hedef veri alanının yinelemeli bir alt kümesidir.

Sürekli olarak benzer yaprak içi (veya düğüm içi) hedef değerleri ve ağacın herhangi bir seviyesinde kademeli olarak farklı yaprak arası (veya düğümler arası değerler) içerir. Bir tür karar ağacı olan 'porfir ağacı' bilinen en eski sınıflandırma ağacı diyagramıdır ve bir zamanlar Yunan filozofu tarafından kullanılmıştır.

3. yüzyılda porfir (Taş) Karar ağaçlarının bu erken hesaplama öncesi kökenleri, karar ağaçlarının hem sezgisel hem de güçlü görsel metaforlar olan bağlamsal olarak açığa çıkaran görselleri yansıtma ve kapsülleme konusunda kalıcı olarak yararlı, doğuştan gelen bir yeteneğini doğrular. 20. yüzyıla doğru ilerlersek karar ağaçlarının, yapay zeka ve istatistiksel hesaplamaların yeni ortaya çıkan alanları ile aynı zamanda ortaya çıktığını görürüz.

Decision trees (Karar ağaçı)

Karar Ağacı nedir?

Bir hikaye ile başlayalım. Bir işiniz olduğunu ve yeni müşteriler edinmek istediğinizi varsayalım. Ayrıca sınırlı bir bütçeniz var ve reklamcılıkta, dönüştürülme olasılığı en yüksek olan müşterilere odaklandığınızdan emin olmak istiyorsunuz.

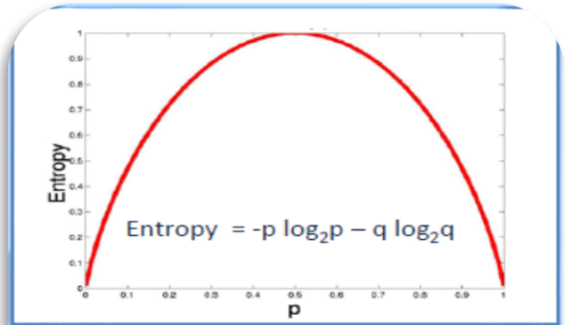
Bu insanların kim olduğunu nasıl anlarsın? Bu müşterileri tanımlayabilecek bir sınıflandırma algoritmasına ihtiyacınız vardır ve kullanışlı olabilecek belirli bir sınıflandırma algoritması karar ağacıdır. Bir karar ağacı, eğitildikten sonra, dönüştürülüp dönüştürülmeyeceğini belirlemek için her yeni müşterinin özelliklerini değerlendirmek için bir dizi kriter verir.

Başlamak için, mevcut müşterilerinizde bulunan verileri kullanarak bir karar ağacı oluşturabilirsiniz. Verileriniz tüm müşterileri, açıklayıcı özelliklerini ve dönüştürülüp dönüştürülmediklerini gösteren bir etiketi içermelidir.

Bir karar ağacı fikri, bir etiketin altına düşen veri noktalarını içeren yeterince küçük bir kümeye ulaşıncaya kadar veri kümesini açıklayıcı özelliklere göre daha küçük veri kümelerine bölmektir.

Veri kümesinin her bir özelliği bir kök [ana] düğüm haline gelir ve yaprak [alt] düğümleri sonuçları temsil eder. Hangi özelliğin bölüneceğine dair karar, ortaya çıkan entropi azalmasına veya bölünmeden elde edilen bilgi kazanımına dayanılarak verilir.

Karar ağaçları için sınıflandırma sorunları genellikle ikilidir - Doğru veya Yanlış, Erkek veya Kadın. Bununla birlikte, karar ağaçları etiketlerin [0,…, K-1] olduğu çok sınıflı sınıflandırma sorunlarını çözmek için de kullanılabilir veya bu örnek için ['Dönüştürülen müşteri', 'Daha fazla fayda ister misiniz', 'Ne zaman dönüştürülür? komik reklamlar görüyorlar ',' Ürünlerimizi hiç satın almayacak '].



Entropy = $-0.5 \log_2 0.5 - 0.5 \log_2 0.5 = 1$

Decision trees (Karar ağaçı)

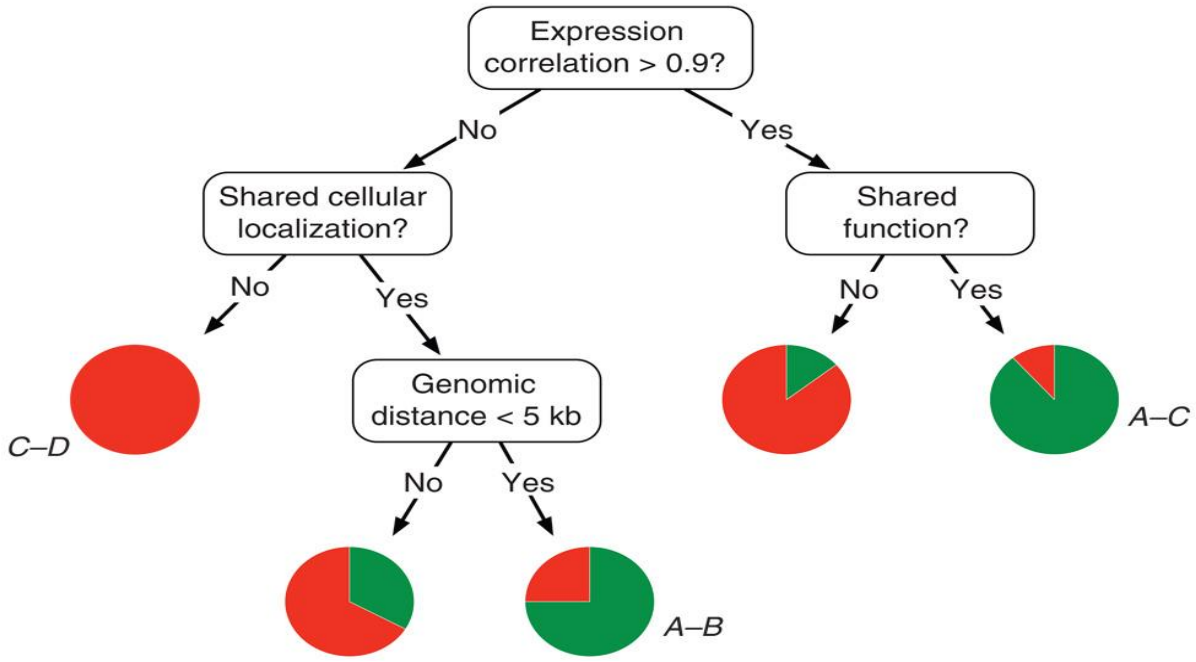
Karar ağaçları, sınıf etiketlerinin bilindiği bir dizi eğitim örneği analiz edilerek oluşturulur. Daha sonra, daha önce görülmemiş örnekleri sınıflandırmak için uygulanırlar. Yüksek kaliteli veriler üzerinde eğitildiyse, karar ağaçları çok doğru tahminlerde bulunabilir.

Şekil 1.

a

Gene Pair	Interact?	Expression correlation	Shared localization?	Shared function?	Genomic distance
A-B	Yes	0.77	Yes	No	1 kb
A-C	Yes	0.91	Yes	Yes	10 kb
C-D	No	0.1	No	No	1 Mb
⋮					

b



Karar ağacının protein-protein etkileşimlerini nasıl tahmin edebileceğine dair varsayımsal bir örnek

(a) Her veri maddesi, çeşitli özelliklerle ilişkili bir gen çiftidir. Bazı özellikler gerçek değerli sayılardır (genler arasındaki kromozomal mesafe veya bir dizi koşul altında ekspresyon profillerinin korelasyon katsayısı gibi). Diğer özellikler kategoriktir (proteinlerin birlikte lokalize olup olmadığı veya aynı fonksiyonla açıklanması gibi). Sadece birkaç eğitim örneği gösterilmiştir.

(b) Her düğümün veri öğelerinin tek bir özelliğini soran bir evet / hayır sorusu içerdiği varsayımsal bir karar ağacı. Soruların cevaplarına göre örnek bir yaprağa ulaşır. Dairesel grafikler, her bir yaprağa ulaşan eğitim örneklerinden gelen (yeşil) ve etkileşmeyenlerin (kırmızı) yüzdelerini gösterir. Yeni örneklerin, ağırlıklı olarak yeşil bir yaprağa ulaşmaları durumunda etkileşime girmesi veya ağırlıklı olarak kırmızı bir yaprağa ulaşmaları halinde etkileşmemesi beklenmektedir. Uygulamada, rastgele proteinler protein-protein etkileşimlerini tahmin etmek için kullanılmıştır15.

Cevaplar verimli bir şekilde hesaplanabildiği sürece ağaçtaki sorular keyfi olarak karmaşık olabilir. Bir sorunun yanıtları, {A, C, G, T} gibi küçük bir kümedeki değerler olabilir. Bu durumda, bir düğümün olası her değer için bir alt öğesi vardır. Birçok durumda, veri öğeleri gerçek değerli özelliklere sahip olacaktır. Bunları sormak için, ağaç "değer> k?" Şeklinde evet / hayır soruları kullanır. yalnızca verilerde meydana gelen değerlerin olası eşikler olarak test edilmesi gereken bazı k eşiği için. Aynı anda birçok özelliğin doğrusal veya mantıksal kombinasyonlarını alarak daha karmaşık sorular kullanmak da mümkündür5.

Karar ağaçları bazen sinir ağları ve destek vektör makineleri gibi diğer sınıflandırıcılardan daha yorumlanabilir olabilir, çünkü verilerle ilgili basit soruları anlaşılır bir şekilde birleştirirler. Karar ağaçlarından karar kurallarının çıkarılmasına yönelik yaklaşımlar da başarılı olmuştur. Ne yazık ki, giriş verilerindeki küçük değişiklikler bazen inşa edilmiş ağaçta büyük değişikliklere yol açabilir. Karar ağaçları, gerçek değerli ve kategorik özelliklerin karışımı olan öğeleri ve bazı eksik özelliklere sahip öğeleri işlemek için yeterince esnektir. Tek bir karar sınırına dayanan (lojistik regresyon veya destek vektör makineleri gibi) sınıflandırıcılar ile kolayca elde edilemeyen verilerin birçok bölümünü modelleyecek kadar etkileyici. Bununla birlikte, bir hiper düzlemle sınıflara mükemmel şekilde bölünebilen veriler bile, sadece basit eşik testleri kullanılırsa büyük bir karar ağacı gerektirebilir. Karar ağaçları doğal olarak ikiden fazla sınıf ile sınıflandırma problemlerini destekler ve regresyon problemlerini ele almak için değiştirilebilir. Son olarak, bir kez inşa edildiğinde, yeni öğeleri hızlı bir şekilde sınıflandırır.

Karar ağaçları ile sınıflandırma

Karar ağaçları, soru seçimine rehberlik etmek için etiketli eğitim örnekleri kullanılarak adım adım soru düğümleri eklenerek büyütülür1,2. İdeal olarak, tek ve basit bir soru, eğitim örneklerini sınıflarına mükemmel bir şekilde ayıracaktır. Böyle mükemmel bir ayrılma sağlayan bir soru yoksa, örnekleri olabildiğince temiz bir şekilde ayıran bir soru seçeriz.

İyi bir soru, heterojen sınıf etiketli bir öğe koleksiyonunu neredeyse homojen etiketli alt kümelere böler, böylece verileri her katmanda çok az fark olacak şekilde katmanlaştırır. Bir dizi maddede homojen olmayanlık veya safsızlık derecesini değerlendirmek için çeşitli önlemler tasarlanmıştır. Karar ağaçları için en yaygın iki önlem entropi ve Gini indeksidir. Varsayalım ki bir dizi eğitim öğesi kullanarak öğeleri m sınıfında sınıflandırmaya çalışıyoruz. Pi (i = 1,…, m), E sınıfının i öğelerinin parçası olsun. Olasılık dağılımının $p_{i=1}^{m}$= 1 entropisi, E setinin safsızlığının makul bir ölçüsünü verir. Entropi, $-\sum_{i=1}^{n} \boldsymbol{P_i log_2(P_i)}$, tek bir pi 1'e eşit olduğunda ve diğerleri 0 olduğunda en düşüktür. tüm pi eşit olduğunda maksimize edilir. Diğer bir safsızlık ölçüsü olan Gini indeksi 2, $1-\sum_{i=1}^{n}(\boldsymbol{p_i^2})$, ile hesaplanır. E kümesi yalnızca bir sınıftan öğeler içerdiğinde, bu yine sıfırdır.

Bir safsızlık ölçüsü I verildiğinde, ortaya çıkan çocuk düğümlerinin safsızlığının ağırlıklı ortalamasını en aza indiren bir soru seçiyoruz. Yani, k olası cevapları olan bir soru E'yi alt kümelere E1…, Ek olarak ayırırsa, simge durumuna küçültmek için bir soru seçeriz $\sum_{i=1}^{k}(|E_j|/|E|)\, I(E_j)$. Birçok durumda, tüm olasılıkları numaralandırarak en iyi soruyu seçebiliriz. Eğer entropi fonksiyonu ise, o zaman ana düğümdeki sınıfların dağılımının entropisi ile çocukların entropisinin bu ağırlıklı ortalaması arasındaki farka bilgi kazancı denir. Kullback-Leibler sapması6 ile ifade edilebilen bilgi kazancının her zaman negatif olmayan bir değeri vardır.

Karar ağaçları ve diğer varyantların toplulukları.

Tek karar ağaçları mükemmel sınıflandırıcılar olabilse de, çoğu zaman karar ağaçlarının toplanması sonuçları birleştirilerek artan doğruluk elde edilebilir8-10. Karar ağaçlarının toplulukları bazen en iyi performans gösteren sınıflandırıcılar arasındadır3. Rasgele ormanlar ve güçlendirme, karar ağaçlarını birleştirmek için iki stratejidir.

Rastgele ormanlar8 yaklaşımında, birçok farklı karar ağacı rastgele bir ağaç oluşturma algoritması ile yetiştirilir. Eğitim seti, orijinaline eşit boyutta değiştirilmiş bir eğitim seti üretmek için değiştirilerek örneklenir, ancak bazı eğitim öğeleri birden fazla kez dahil edilir. Ayrıca, her düğümde soru seçerken, özelliklerin sadece küçük, rastgele bir alt kümesi dikkate alınır. Bu iki değişiklikle, her çalışma biraz farklı bir ağaçla sonuçlanabilir. Ortaya çıkan karar ağaçları topluluğunun tahminleri en yaygın tahmin alınarak birleştirilir. Tek bir ağaca bağlı kalmak yerine iyi bir hipotez koleksiyonunu sürdürmek, yeni bir örneğin ağaçların çoğu tarafından yanlış sınıfa atanarak yanlış sınıflandırma olasılığını azaltır.

Boosting10, en problematik olana odaklanmak için eğitim örneklerini tekrar tekrar yeniden ağırlıklandırarak çoklu sınıflandırıcıları daha güçlü bir sınıflandırıcıda birleştirmek için kullanılan bir makine öğrenme yöntemidir. Uygulamada, karar ağaçlarını birleştirmek için çoğu kez yükseltme uygulanır. Alternatif karar ağaçları11, tek bir sorudan oluşan karar ağaçları olan karar kütüklerine dayalı zayıf sınıflandırıcıları birleştirmek için bir çeşit yükseltmenin uygulanmasından kaynaklanan karar ağaçlarının genelleştirilmesidir. Alternatif karar ağaçlarında, ağacın seviyeleri standart soru düğümleri ile ağırlık içeren ve keyfi olarak çocuk sahibi olan düğümler arasında değişir. Standart karar ağaçlarının aksine, öğeler birden fazla yol alabilir ve yolların karşılaştığı ağırlıklara göre sınıflara atanır. Alternatif karar ağaçları, doğrudan standart karar ağaçlarına takviye uygulamaktan elde edilenlerden daha küçük ve daha yorumlanabilir sınıflandırıcılar üretebilir.

Hesaplamalı biyolojiye uygulamalar:

Karar ağaçları, doğru tahminler yapmak için çeşitli veri türlerini bir araya getirmedeki yararlılıkları nedeniyle hesaplama biyolojisi ve biyoinformatik içinde geniş uygulama alanı bulmuştur. Burada, kullanımlarının birçok örneğinden sadece birkaçından bahsediyoruz.

A ve B genleri arasındaki sentetik hasta ve ölümcül (SSL) genetik etkileşimler, organizma hem A hem de B çıkarıldığında zayıf büyüme (veya ölüm) gösterdiğinde ortaya çıkar, ancak A veya B ayrı ayrı devre dışı bırakıldığında gerçekleşmez. Wong ve ark.12, Saccharomyces cerevisiae'deki SSL etkileşimlerini, iki proteinin fiziksel olarak etkileşime girip girmediği, hücrede aynı yere lokalize olup olmadığı veya bir veritabanında kaydedildiği gibi çeşitli özellikleri kullanarak karar ağaçları uyguladı. Düşük yanlış pozitif oranıyla yüksek oranda SSL etkileşimi belirleyebildiler. Ayrıca, hesaplanan ağaçların analizi SSL etkileşimlerinin altında yatan çeşitli mekanizmalara işaret etti.

Hesaplamalı gen bulucuları, ökaryotik genlerin doğru eksonintron yapısını belirlemek için çeşitli yaklaşımlar kullanır. Ab initio gen bulucuları, diziye özgü bilgileri kullanırken, hizalamaya dayalı yöntemler, ilgili türler arasında dizi benzerliğini kullanır. Allen ve ark.13, JIGSAW sistemindeki karar ağaçlarını birçok farklı gen bulma yönteminden elde edilen kanıtları birleştirmek için kullandı, bu da insan genomundaki genleri ve diğer türlerin genomlarını bulmak için mevcut en iyi yollardan biri olan entegre bir yöntemle sonuçlandı.

Middendorf ve ark.14, bir S. cerevisiae geninin, düzenleyici bölgesinin sekansı verildiğinde belirli transkripsiyon regülatörü ekspresyonu koşulları altında yukarı veya aşağı regüle edileceğini tahmin etmek için alternatif karar ağaçları kullanmıştır. Hedef genlerin ekspresyon durumunu tahmin eden iyi performansa ek olarak, hedef genlerin ekspresyonunu kontrol ettiği görülen motifleri ve düzenleyicileri tanımlayabildiler.

Overfitting(Aşırı Uyum)

- Overfitting karar ağacı modelleri ve diğer pek çok tahmin modeli için önemli bir sorundur. Öğrenme algoritması etkileyecek şekilde eğitim seti hatalarını azaltmaya devam edildiğinde overfitting olur. Bir karar ağaç inşasında overfitting'ten kaçınmak için genelde iki yaklaşım kullanılır.

- Pre-pruning: Sınırlandırma işleminde önce ağacın büyümesini durdurmak.

- Post-pruning: öncelikle tüm ağacı oluşturup daha sonra ağaçtaki gereksiz kısımları çıkarmak.

- Uygulamada ne zaman pruning (budama) işleminin yapılacağını belirlemedeki zorluk sebebiyle ilk yaklaşım pek kullanılmaz. İkinci yaklaşım çok daha başarılıdır. Bu yaklaşım aşağıdaki adımlara dikkat edilmelidir:

- Budama işlemine karar vermek için eğitim verisinden farklı bir veri seti kullanmak. Bu veri setine doğrulama veri seti (validation dataset) denir. Validation dataset gereksiz düğümlere karar vermek için kullanılır.

- Bir karar ağacı elde ettikten sonra, hata tahmini (error estimation) ve önem testi (Significance testing – Chi Square Testing) gibi istatiksel metotlar kullanarak eğitim verisi üzerinde budama ve genişleme (expanding – ağaça yeni node'lar ekleme) olup olmayacağına karar verilir.

- Minimum Description Length principle: karar ağacı ile eğitim veri seti arasında bir ölçüdür. Boyut(tree) + Boyut(sınıflanamayan(tree)) minimize olduğunda ağaç büyümesini durdurma.

Avantajları :

- Anlaması Kolay: Analitik olmayan geçmişe sahip insanlar için bile karar ağacı çıktısını anlamak çok kolaydır. Bunları okumak ve yorumlamak için herhangi bir istatistiksel bilgi gerektirmez.
-
- Veri keşfinde yararlı: Karar ağacı, en önemli değişkenleri ve iki veya daha fazla değişken arasındaki ilişkiyi tanımlamanın en hızlı yollarından biridir.
- Karar ağaçları dolaylı olarak özellik seçimi yapar.
- Karar ağaçları veri hazırlama için kullanıcılardan nispeten az çaba gerektirir.
- Daha az veri temizliği gerekir
- Veri türü bir kısıtlama değildir: Hem sayısal hem de kategorik değişkenleri işleyebilir. Ayrıca çoklu çıktı sorunlarını da çözebilir.
- Parametrik Olmayan Yöntem: Karar ağacı parametrik olmayan bir yöntem olarak kabul edilir.Bu, karar ağaçlarının uzay dağılımı ve sınıflandırıcı yapısı hakkında hiçbir varsayımları olmadığı anlamına gelir.
- Doğrusal olmayan: parametreler arasındaki ilişkiler ağaç performansını etkilemez.
- Ayarlanacak hiper parametre sayısı neredeyse sıfırdır.

Dezavantajları :

- Aşırı uyum: Karar ağacı öğrenicileri, verileri iyi bir şekilde genelleştirmeyen aşırı karmaşık ağaçlar oluşturabilir. Bu sorun, model parametreleri ve budama üzerindeki kısıtlamaları ayarlayarak çözülür.

- Sürekli değişkenler için uygun değil: Sürekli sayısal değişkenlerle çalışırken, karar ağacı farklı kategorilerdeki değişkenleri kategorilere ayırdığında bilgileri kaybeder.

- Karar ağaçları kararsız olabilir, çünkü verilerdeki küçük değişiklikler tamamen farklı bir ağacın üretilmesine neden olabilir. Buna, torbalama ve artırma gibi yöntemlerle düşürülmesi gereken varyans denir.

- Açgözlü algoritmalar, küresel olarak en uygun karar ağacını döndürmeyi garanti edemez. Bu, özelliklerin ve örneklerin değiştirme ile rastgele örneklendiği birden fazla ağaç eğitilerek hafifletilebilir.

- Karar ağacı öğrenenleri bazı sınıfların baskın olması durumunda önyargılı ağaçlar oluşturur. Bu nedenle karar ağacına uymadan önce veri kümesinin dengelenmesi tavsiye edilir.

- Kategorik değişkenlere sahip bir karar ağacında bilgi kazanımı, daha büyük olan öznitelikler için taraflı bir yanıt verir. kategorileri.

- Genellikle, bir veri kümesi için diğer makine öğrenme algoritmalarına kıyasla düşük tahmin doğruluğu verir.

- Birçok sınıf etiketi olduğunda hesaplamalar karmaşık hale gelebilir.

Budama (Pruning)

Karar ağaçlarında aşırı sıkışmayı ve ağaç derinliğini azaltmak için bir yöntemdir,daha kolay anlaşılırlar ve genellikle daha hızlıdırlar.İki tür budama vardır: erken budama ve geç budama.

Erken budama(Pre-pruning)

Ağacın büyümesini erken durduran yaklaşımlar Bir düğüme ulaşan örnek sayısı, eğitim verilerinin belirli bir yüzdesinden daha küçükse o düğüm artık bölünmez.

Az sayıda örneğe bağlı olarak alınan kararlar genelleme hatasını artırır.

Daha hızlı çözüm.

geç budama (Post-pruning)

Karar ağaçlarında uygulanan "greedy" algoritması: her adımda bir düğüm ekler, geriye dönüp başka bir seçenek düşünmez. Bu durumun tek istinası: gereksiz alt ağaçların bulunup budanmasıdır.

Daha doğru çözüm.

- geç budama modelin aşırı takılmasına neden olabilir.
- geç budama şu anda Python'un scikit kütüphanesi mevcut değildir, ancak R'de mevcuttur.

Topluluklar

Topluluk oluşturmak, farklı modellerin sonuçlarının toplanmasını içerir. Toplama karar ağaçları torbalama ve rastgele ormanlarda kullanılırken, topluluk regresyon ağaçları güçlendirme için kullanılır.

Torbalama(Bagging) / Bootstrap toplama

Torbalama, her biri verilerin farklı bir önyükleme örneği üzerinde eğitilmiş birden fazla karar ağacı oluşturmayı içerir. Önyükleme değiştirme ile örnekleme içerdiğinden, örnekteki verilerin bir kısmı her ağacın dışında bırakılır.

Sonuç olarak, oluşturulan karar ağaçları, eğitim örneğine aşırı uyum sorununu çözen farklı örnekler kullanılarak yapılır. Karar ağaçlarının bu şekilde birleştirilmesi toplam hatayı azaltmaya yardımcı olur, çünkü topluluğun önyargısında bir artış olmadan eklenen her yeni ağaç ile modelin varyansı azalmaya devam eder.

Rastgele Orman (Random Forest)

Alt uzay örneklemesi kullanan bir torba karar ağacı rastgele bir orman olarak adlandırılır. Ormandaki ağaçları süsleyen her düğüm bölmesinde sadece özelliklerin bir seçimi göz önünde bulundurulur.

Rasgele ormanların bir diğer avantajı, yerleşik bir doğrulama mekanizmasına sahip olmalarıdır. Her model için verilerin yalnızca bir yüzdesi kullanıldığından, modelin performansına ilişkin stokta kalma hatası, her modelin dışında kalan örneğin %37'si kullanılarak hesaplanabilir.

Arttırma

Artırmak, güçlü bir yordayıcı oluşturmak için zayıf öğrenenlerin (regresyon ağaçları) bir araya getirilmesini içerir. Artırılmış bir model zaman içinde, önceki öğrenenler tarafından hatayı en aza indiren yeni bir ağaç eklenerek oluşturulur. Bu, yeni ağacın önceki ağaçların kalıntılarına yerleştirilmesiyle yapılır.

Şimdiye kadar açık değilse, birçok gerçek dünya uygulaması için, tek bir karar ağacı tercih edilen bir sınıflandırma değildir, çünkü yeni örneklere çok zayıf ve genelleme olasılığı yüksektir. Bununla birlikte, bir dizi karar veya regresyon ağacı, aşırı uyum dezavantajını en aza indirir ve bu modeller yıldız, son teknoloji sınıflandırma ve regresyon algoritmaları haline gelir.

SONUÇLAR

Çok yönlü temaların birçok varyasyonu vardır: otonom ve seri örnekler; satır ve sütun yeniden ağırlıklandırma şemaları; değiştirme örnekleri vs değiştirmesiz örnekler; ve bunun gibi. En iyi tahmin, karar ağaçları üzerinde yapılan iyileştirmeler, birçok çoklu yöntemde gösterilmiştir. Eğitim verilerinin boyutu artmaya devam ettikçe, birden fazla otonom ağacın yaklaşımında bariz faydalar vardır, çünkü bu ağaçlar, toplam bir etki üretilmeden önce bağımsız olarak paralel olarak hesaplanabilir. İlk eğitim verilerinin boyutu arttıkça, değiştirilmeden örneklemeye de vurgu yapılmaktadır. Daha büyük eğitim verileriyle, değiştirilmeden örnekleme, model sonuçlarındaki farklılıkların benimsenmesini pekiştirme eğilimindedir. Bu, artık çok yönlü yöntemlerin potansiyel gücü olarak kabul edilmektedir.Bugüne kadar, birçok çok yönlü yöntem çeşitli durumlarda güçlü yanlar göstermektedir. Bu alan geliştikçe hangi yöntemin hangi koşullarda en iyi olduğu netleşebilir. Bu alandaki yeniliğin hızı göz önüne alındığında, gelişmiş yöntemlerin ve yeni paradigmaların ortaya çıkmaya devam etmesi muhtemeldir.

Referanslar

1. Quinlan JR. C4.5: Programs for Machine Learning. San Mateo, CA, USA: Morgan Kaufmann Publishers; 1993.

2. Chen X-W, Liu W. Prediction of protein-protein interactions using random decision forest framework. Bioinformatics. 2005.

3. Caruana R, Niculescu-Mizil A. An empirical comparison of supervised learning algorithms. In: Cohen WW, Moore A, editors. Machine Learning, Proceedings of the Twenty-Third International Conference; New York: ACM; 2003.

4. Murthy SK, Kasif S, Salzberg S. A system for induction of oblique decision trees. J. Artif. Intell. Res. 1994

5. MacKay DJC. Information Theory, Inference and Learning Algorithms. Cambridge, UK: Cambridge University Press; 2003.

6. Breiman L. Random forests. Mach. Learn. 2001

Daha Fazla Okuma

de Ville B, Neville P. Decision Trees for Analytics Using SAS Enterprise Miner. Cary, NC: SAS Press; 2013.