Karar Ağaçları

M. Sinan İyisoy

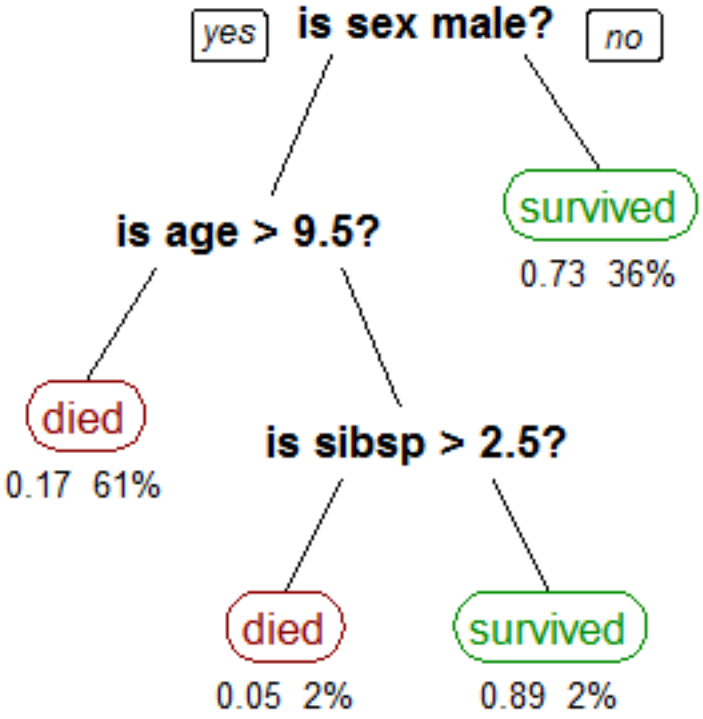
Karar Ağacı

- Bir karar ağacı karar vermeyi desteklemek amacıyla oluşturulmuş ve karar verme algoritmasını gösteren bir yapıdır.
- Karar ağacında dallar ve bu dallara atanmış olasılık değerleri vardır.

Karar Ağacı

- Titanik yolcuları için hayatta kalma karar ağacı

sibsp = gemideki kardeş ya da eş sayısı



Karar Ağaçları ile Öğrenme

- Karar ağaçlarının bir tahmin modeli olarak kullanıldığı bu öğrenme modeli istatistik, makine öğrenmesinde ve veri madenciliğinde sıklıkla kullanılır.
- Bir bağımlı ve birden çok sayıda bağımsız değişken vardır.
- Bağımlı değişkenin kategorik olduğu durumda classification (sınıflama) ağacı, sürekli olduğu durumda regression ağacı olarak adlandırılır.

Karar Ağaçlarının Avantajları

- Kendiliğinden değişken seçimi ve taraması yaparak en uygun değişkenleri seçer.
- Kullanıcının veri üzerinde düzenleme yapması çoğu zaman gerekli değildir. Standartlaştırma gerektirmez, eksik verileri iyi tolere eder.
- Değişkenler arasındaki doğrusal olmayan ilişkilerden etkilenmez. Doğrusallık varsayımı yoktur.
- Yorumlaması çok kolaydır.

CART

- Classification and Regression Trees karar ağaçları için genel bir addır. İlk olarak Breiman tarafından ortaya atılmıştır.
- Karar ağaçları için kullanılan değişik yöntemler vardır. Bu yöntemlerle birden çok karar ağacı üretilir. Bu ağaçlar birlikte değerlendirilerek sonuç üretilir.

Yöntemler

- Bagging: train (eğitim) verisinden yerine koyarak çekilen veri ile değişik karar ağaçları elde edilir. Bu şekilde üretilmiş birden çok train verisi kullanır. (Bootstrap aggregation)
- Random forest: sınıflama oranını arttırmak için birden çok karar ağacı kullanır. Aynı zamanda rastgele elde edilmiş bir bağımsız değişken kümesini kullanır. Elde edilen tahminlerin ortalamasını ya da modunu sonuç olarak verir.

Algoritmalar

- CART için geliştirilmiş değişik algoritmalar vardır.
- ID3: C4.5 algoritmasının atası olan bu algoritma Rose Quinlan tarafından 1986'da geliştirilmiştir. Karar vermekte entropiyi kullanır.
- C4.5: Yine Quinlan tarafından 1993 yılında geliştirilmiş bir algoritmadır. Entropiyi kullanır.

Algoritmalar

- CHAID: 1980 yılında Gordon V.Kass tarafından geliştirilen bu algoritma Bonferroni düzetmeli anlamlılık testi yaparak çıkarımda bulunur. **CH**i-squared **A**utomatic **I**nteraction **D**etection
- Pratikte pazarlama alanında sıklıkla kullanılan bu yöntemle müşteri grupları seçilir ve müşteriler davranış biçimlerinin başka değişkenleri nasıl etkilediği tahmin edilir.

Algoritmalar

- MARS: Multivariate Adaptive Regression Splines 1991 yılında Jerome Friedman tarafından geliştirilmiş nonparametrik bir regresyon yöntemidir.
- Regresyon modellerinden daha esnektir. Anlaması ve yorumlaması daha kolaydır.

Algoritmalar

- CART: Aynı isimli algoritma.
- CRUISE: Hyunjoong Kim ve Wei-Yin Loh tarafından sınıflama için geliştirilmiş bir algoritmadır.
- QUEST: Wei-Yin Loh ve Yu-Shan Sih tarafından geliştirilmiş iki parçalı bir sınıflama algoritmasıdır.
- GUIDE: Wei-Yin Loh tarafından geliştirilmiş bir sınıflama ve regresyon ağacı algoritmasıdır.

Karşılaştırma

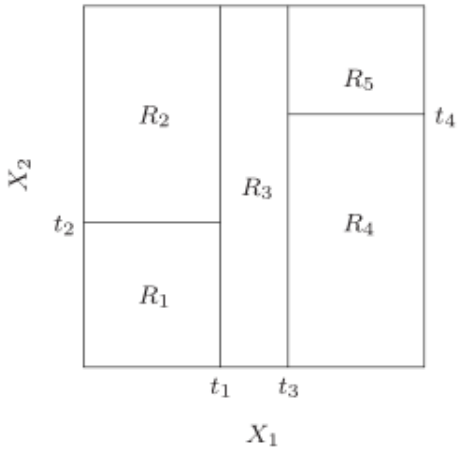
Comparison of GUIDE, QUEST, CRUISE, CART, and C4.5 classification tree algorithms

	GUIDE	QUEST	CRUISE	CART	C4.5
Unbiased splits	Yes	Yes	Yes	No	No
Splits per node	2	2	≥ 2	2	2
Interaction detection	Yes	No	Yes	No	No
Importance ranking	Yes	No	No	Yes	No
Class priors	Yes	Yes	Yes	Yes	No
Misclassification costs	Yes	Yes	Yes	Yes	No
Linear splits	Yes	Yes	Yes	Yes	No
Categorical splits	Subsets	Subsets	Subsets	Subsets	Atoms
Node models	S, K, N	S	S, L	S	S
Missing values	Special	Imputation	Surrogate	Surrogate	Weights
Tree diagrams	Text and LaTeX			Proprietary	Text
Bagging	Yes	No	No	No	No
Forests	Yes	No	No	No	No

Node models: S = simple, K = kernel, L = linear discriminant, N = nearest-neighbor.

Regresyon Ağaçları

- Sürekli bir bağımlı değişken Y ve iki adet sürekli bağımsız değişkenin X_1 ve X_2 olduğu bir regresyon modelini düşünelim.
- Feature space (özellik uzayı) değişik bölgelere ayrılır. Recursive binary splitting

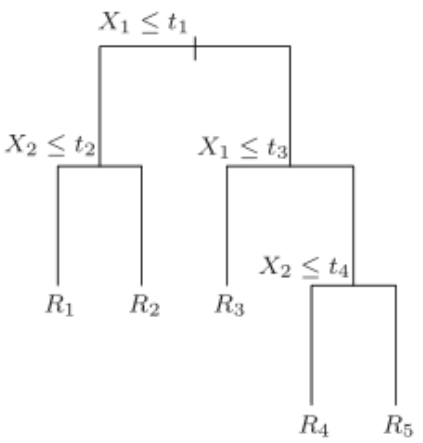


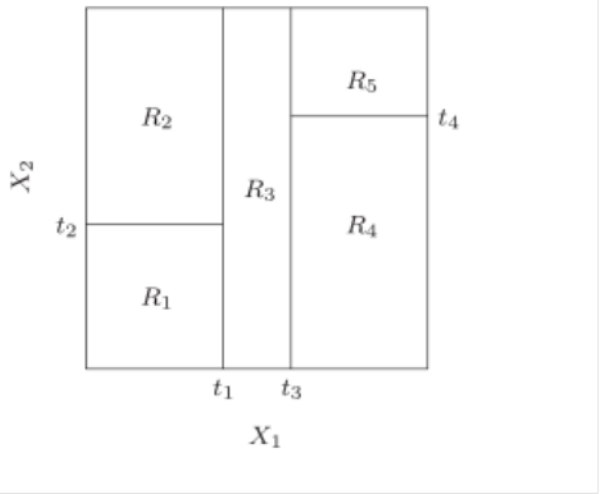
Regresyon Ağaçları

- Oluşan bu 5 bölge R_m için birleşik bir regresyon modeli yazılır.

$$\hat{f}(X) = \sum_{m=1}^{5} c_m I\{(X_1, X_2) \in R_m\}.$$

- Bu model aşağıdaki ağaç ile gösterilir. Sol taraf Evet, Sağ taraf Hayır.





Ağacın Büyümesi

- p bağımsız değişkenli ve bir cevap değişkenli modelimizde N adet gözlem olsun.
- M adet bölgemiz olsun R_1,R_2,.. R_m. Modelimiz her bölge için sabit bir sonuç c_m üretsin. Bu durumda $f(x) = \sum_{m=1}^{M} c_m I(x \in R_m).$ olur.
- Kriter olarak kareler toplamını $\sum(y_i - f(x_i))^2$ minimize etmeyi alırsak, c_m olarak $\hat{c}_m = \text{ave}(y_i | x_i \in R_m).$ seçmek uygundur.

Ağacın Büyümesi

- Kareler toplamını minimize eden bölge ayrımını bulmak hesaplama açısından zordur.
- j ayırma değişkeni ve s ayırma değeri olarak alındığında aşağıdaki 2 bölge edilir.

$$R_1(j,s) = \{X|X_j \leq s\} \text{ and } R_2(j,s) = \{X|X_j > s\}.$$

- Aşağıdaki şartı sağlayan j ve s yi ararız.

$$\min_{j,\,s}\Big[\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_2(j,s)}(y_i-c_2)^2\Big]$$

Ağacın Büyümesi

- Parantez içini minimize eden değerler

$$\hat{c}_1 = \text{ave}(y_i | x_i \in R_1(j,s)) \text{ and } \hat{c}_2 = \text{ave}(y_i | x_i \in R_2(j,s)).$$

- Tüm değerler taranarak uygun j ve s değerleri bulunur. Bulunan bu değerlere göre veri iki parçaya bölünür ve daha sonra aynı işlem bu iki parça için devam eder.
- Buradaki önemli bir soru bu işlemin nereye kadar süreceğidir?

Ağacın Büyümesi

- Burada tercih edilen bir strateji T_0 ağacını büyütmek için belli mininum node sayısına ulaşınca durmaktır.
- Cost-complexity prunning
- $T \subset T_0$ şeklinde bir alt ağaç olsun. Bu ağaç T_0 ağacının bazı nodlarını kapatarak elde edilmiştir. Terminal nodlar m ile indekslensin ve ilgili bölge R_m olsun.

Ağacın Büyümesi

$$N_m = \#\{x_i \in R_m\},$$
$$\hat{c}_m = \frac{1}{N_m} \sum_{x_i \in R_m} y_i,$$
$$Q_m(T) = \frac{1}{N_m} \sum_{x_i \in R_m} (y_i - \hat{c}_m)^2,$$

- şeklinde tanımlanınca cost complexity criterion aşağıdaki gibi olur

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha |T|.$$

- Burada her α değerine karşılık C_α (T) yi minimize etmek için T_α alt ağacını bulmak gereklidir.

Ağacın Büyümesi

- α bir tuning parametresi olarak ağacın büyüklüğü ile veriye uyumu arasındaki dengeyi sağlar. Büyük α değerleri küçük ağaçlara, küçük α değerleri büyük ağaçlar üretir.
- α nın kestirimi 5- ya da 10- kesimli çapraz geçerlilik ile kareler toplamının minimize edilmesi ile yapılır. Sonuçta elde edilen ağacımız $T_{\hat{\alpha}}$ dır.

Sınıflama Ağaçları

- Cevap değişkeni kategorik olduğunda algoritmada bazı değişiklikler yapılması gerekir. Regresyon ağacında impurity measure

$$Q_m(T) = \frac{1}{N_m} \sum_{x_i \in R_m} (y_i - \hat{c}_m)^2$$

olarak tanımlanmıştı. Şimdi ise

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k),$$

kullanılır.

Sınıflama Ağaçları

- m nodundaki gözlemleri $k(m) = \arg\max_k \hat{p}_{mk}$ sınıfına atarız.
- Başka impurity measure lar da vardır.

Misclassification error:	$\frac{1}{N_m} \sum_{i \in R_m} I(y_i \neq k(m)) = 1 - \hat{p}_{mk(m)}$.
Gini index:	$\sum_{k \neq k'} \hat{p}_{mk} \hat{p}_{mk'} = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$.
Cross-entropy or deviance:	$-\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$.

- Bu üç ölçü de birbirine benzerdir. Ama Gini index ve cross-entropy ölçüsü türevlenebilir olduğu için nümerik optimizasyona daha uygundur.