

Machine Learning

By: Dr. Omid Mahdi Ebadati

Ph.D. (Computer Science), Delhi



Kharazmi University

Kharazmi University

یادگیری از طریق

درخت تصمیم

Decision Tree

مقدمه

- در یک مسئله یادگیری با دو جنبه مختلف روبرو هستیم:
- نحوه نمایش فرضیه ها
- روشی که برای یادگیری برمی گزینیم
- در این فصل برای نمایش فرضیه ها از درخت تصمیم استفاده میکنیم و برای یادگرفتن این درخت از روش ID3 استفاده میکنیم.

مقدمه (ادامه)

یادگیری درخت تصمیم روشی برای تقریب توابع هدف با مقادیر گسسته است. این روش نسبت به نویز داده ها مقاوم بوده و قادر است ترکیب فصلی گزاره های عطفی را یاد بگیرد.

این روش جزو مشهورترین الگوریتمهای یادگیری استقرائی است که بصورت موفقیت آمیزی در کاربردهای مختلف بکار گرفته شده است.

مقدمه (ادامه)

درخت تصمیم درختی است که در آن نمونه ها را به نحوی دسته بندی میکند که از ریشه به سمت پائین رشد میکنند و یک مدل برای یادگیری نظارت شده است که در نهایت به گره های برگ میرسد

علت نامگذاری آن با درخت تصمیم این است که این درخت فرایند تصمیم گیری را برای تعیین دسته یک مثال ورودی نشان میدهد.

درختها درهوش مصنوعی برای نمایش مفاهیم مختلفی نظیر ساختار جملات، معادلات، حالات بازی، و نیز برای تشخیص بیماریها - دسته بندی گیاهان و استراتژی بازاریابی مشتری استفاده میشود

Introduction

- o A **decision tree** is a decision support tool that uses a **tree-like graph** or model of decisions and their **possible consequences**, including **chance event** outcomes, **resource costs**, and **utility**. It is one way to display an algorithm.

Introduction (Cont.)

- o **Decision Tree is a flow-chart like structure in which a. Internal node represents test on an attribute, b. Each branch represents outcome of test and c. Each leaf node represents class label (decision taken after computing all attributes).** A path from root to leaf represents classification rules.

اجزاء درخت تصمیم

Attribute, Branch, Leaf Node

هر گره داخلی یا غیر برگ (non leaf) با یک ویژگی (attribute) مشخص میشود. این ویژگی سوالی را در رابطه با مثال ورودی مطرح میکند. برای یافتن بهترین صفت در هر گره، در نظر گرفتن یک زیرمجموعه کوچک از نمونه های آموزشی که از آن گره عبور می کنند کافی است.

اجزاء درخت تصمیم

• در هر گره داخلی به تعداد **جوابهای ممکن** با این سوال **شاخه** (branch) وجود دارد که هر یک با مقدار آن جواب مشخص میشوند. که به یک ورودی معین در هر گره و یک آزمون اعمال می شود و یکی از شاخه های وابسته به این گره حاصل می شود.

• **برگ** های این درخت (leaf Node) با **یک کلاس و یا یک دسته از جوابها** مشخص میشوند. که مقدار نوشته شده در برگ خروجی را تشکیل می دهد.

Decision Tree Algorithms

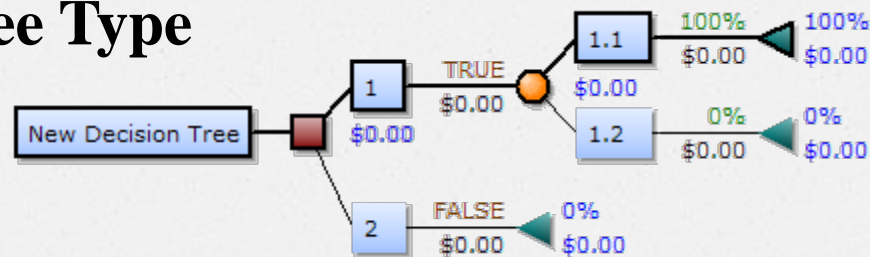
- o There are many specific decision-tree algorithms. Notable ones include:
 - o **ID3** (Iterative Dichotomiser 3)
 - o **C4.5** (successor of ID3)
 - o **CART** (Classification And Regression Tree)
 - o **CHAID** (CHi-squared Automatic Interaction Detector). Performs multi-level splits when computing classification trees.
 - o **MARS**: extends decision trees to better handle numerical data.
- o ID3 and CART were invented independently at around same time (b/w 1970-80), yet follow a similar approach for learning decision tree from training tuples.

Node Types in DT

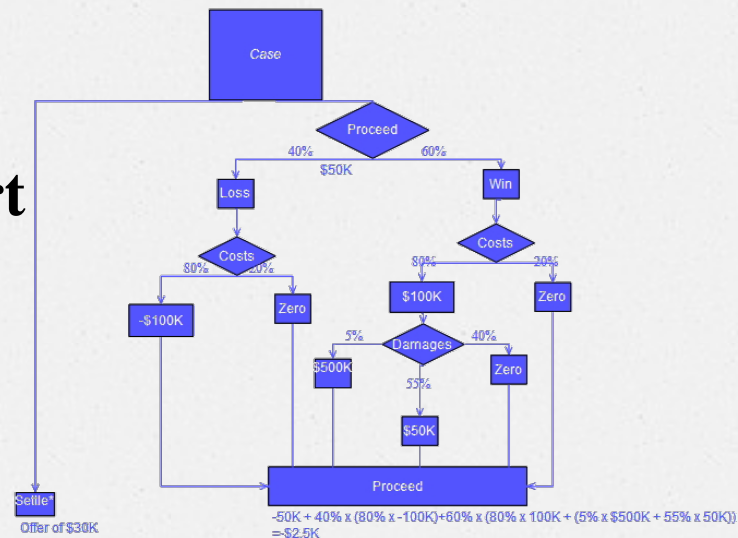
- o A decision tree consists of 3 types of nodes:
 - o **Decision nodes** - commonly represented by **squares**
 - o **Chance nodes** - represented by **circles**
 - o **End nodes** - represented by **triangles**

DT Presentation

Tree Type



Flowchart



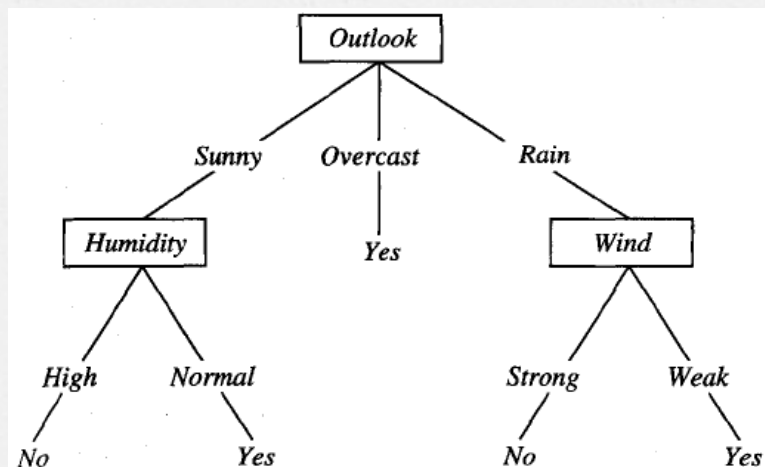
DT (Cont.)

- It is one of the **predictive modeling approaches** used in **statistics, data mining** and **machine learning**.
- Descriptive names for such tree **models** are:
 - **Classification trees**
 - **Regression trees.**
- In these tree structures, **leaves represent class labels** and **branches represent conjunctions of features that lead to those class labels.**

نحوه ساختن درخت

- برای ساختن درخت تصمیم از مثالهایی استفاده میشود که علامت گذاری (label) شده باشند.
- در واقع ورودی سیستم یادگیر مجموعه ای از مثالهاست که هر مثال توسط مجموعه ای از ویژگی ها بیان شده است، هر ویژگی می تواند دارای مجموعه متناهی از مقادیر مختلف باشد. برای هر مثال علاوه بر ویژگیها مقدار دسته بندی آن نیز لازم می باشد.
- در این فصل با درختهای تصمیمی آشنا خواهیم شد که برای **دسته بندی** بولی بکار می روند ولی در حالت کلی می توان یک درخت تصمیم ساخت که برای هر نوع دسته بندی بکار می رود.

نحوه نمایش درخت تصمیم



ارتباط مستقیمی بین درخت
تصمیم و نمایش توابع منطقی
وجود دارد. در واقع هر درخت
تصمیم ترکیب فصلی گزاره های
عطفی است

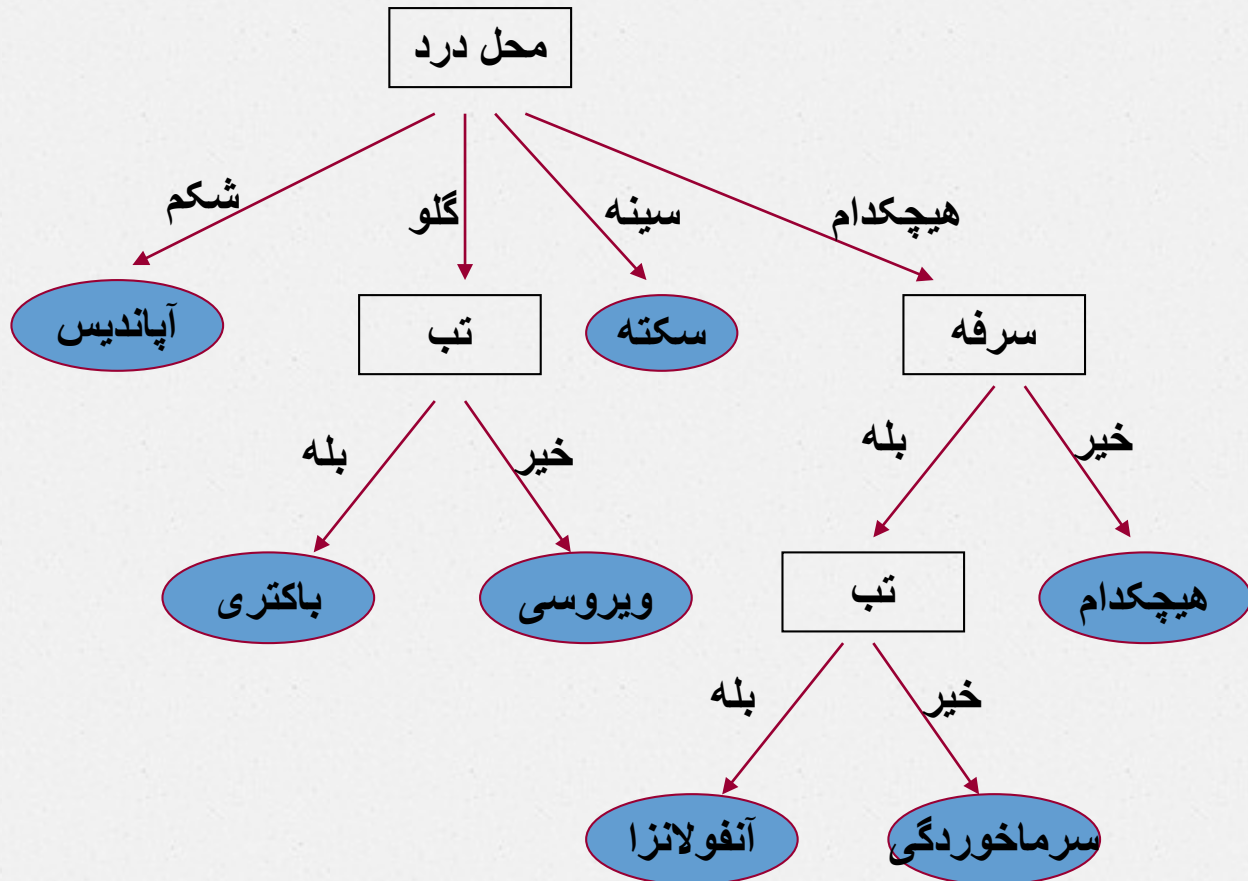
- ($Outlook = Sunny \wedge Humidity = Normal$)
- ∨ ($Outlook = Overcast$)
- ∨ ($Outlook = Rain \wedge Wind = Weak$)

مسیر از ریشه به برگ ترکیب عطفی (AND) از ویژگی ها را مشخص نموده و
خود درخت ترکیب فصلی (OR) این ترکیبات را میسازد.

کاربردها

- درخت تصمیم در مسایلی کاربرد دارد که بتوان آنها را بصورتی مطرح نمود که پاسخ واحدی بصورت نام یک دسته یا کلاس ارائه دهند.
- برای مثال میتوان درخت تصمیمی ساخت که به این سوال پاسخ دهد: بیماری مریض کدام است؟ و یا درختی ساخت که به این سوال پاسخ دهد: آیا مریض به هیپاتیت مبتلاست؟
- برای مسائلی مناسب است که مثالهای آموزشی بصورت زوج (مقدار-ویژگی) مشخص شده باشند.
- تابع هدف دارای خروجی با مقادیر گسسته باشد. مثلا هر مثال با بله و خیر تعیین شود.
- نیاز به توصیف گر فصلی (disjunctive) باشد.

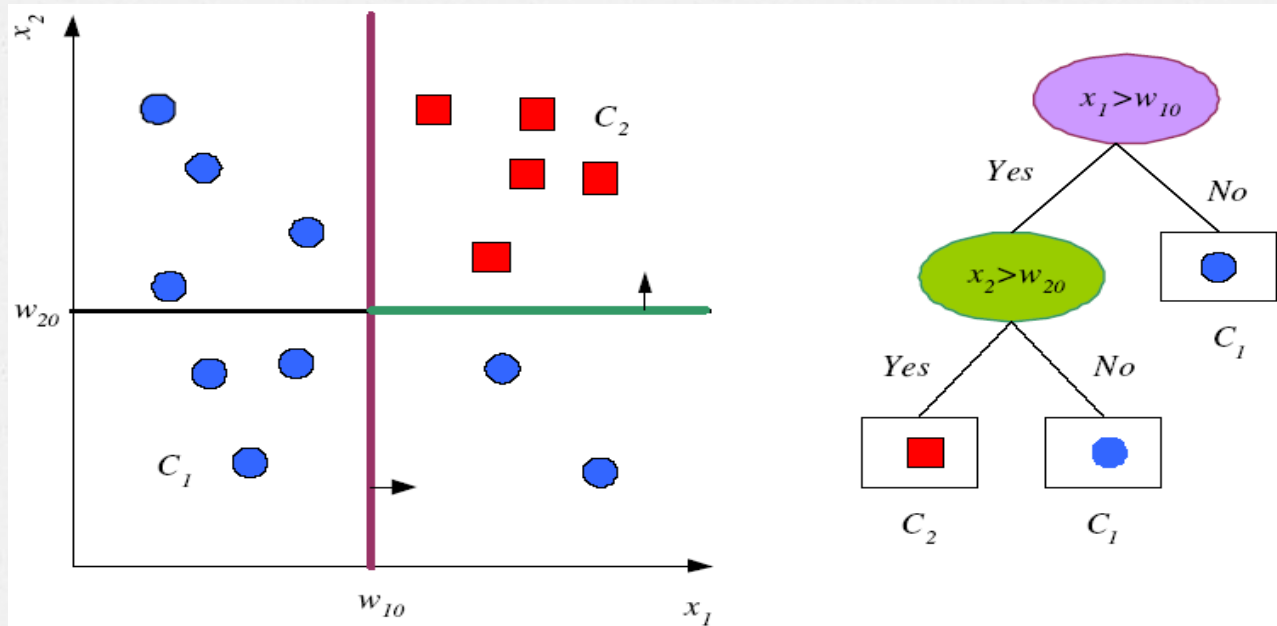
مثال



باتوجه به مثال قبل

- هر برگ این درخت یک کلاس یا دسته را مشخص میکند.
- یک مثال آموزشی در درخت تصمیم به این صورت دسته بندی میشود:
- از ریشه درخت شروع میشود.
- ویژگی معین شده توسط این گره تست می گردد.
- و سپس منطبق با ارزش ویژگی در مثال داده شده در طول شاخه ها حرکت رو به پائین انجام می دهد.
- این فرآیند برای گره های زیردرختان گره جدید تکرار می شود.

مثالی از ترسیم درخت تصمیم از نمونه ها

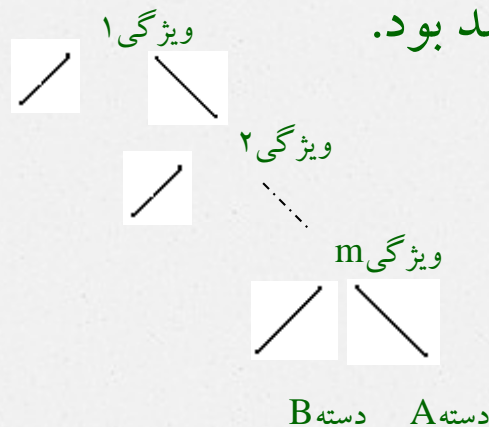


ارتفاع درخت تصمیم

اگر مسئله ما دارای m ویژگی باشد، ارتفاع درخت تصمیم چقدر خواهد بود؟

جواب:

درخت تصمیم دارای **یک ریشه** است که آن خود **یک ویژگی** است، در سؤال از آن ویژگی به **پاسخی** می‌رسیم که آن خود نیز، **ویژگی** است. پس حداکثر ارتفاع درخت m خواهد بود.



Divide and Conquer

- o **Internal decision nodes**
 - o **Univariate:** Uses a **single attribute**, x_j
 - o Numeric x_j : Binary split : $x_j > w_m$
 - o Discrete x_j : n -way split for n possible values
 - o **Multivariate:** Uses all attributes, \mathbf{x}
- o **Leaves**
 - o **Classification:** Class labels, or proportions
 - o **Regression:** Numeric; r average, or local fit

درختهای تک متغیره Univariate (خصوصیات)

در یک درخت تک متغیره، در هر گره داخلی تنها یکی از متغیرها را به کار می برد و تنها متغیرهایی را به کار می برد که لازم و ضروری هستند.

مشخصه های نزدیک به ریشه به طور کلی مهمتر هستند.

برای مثال

$$X_i > W_m$$

X_i نمونه

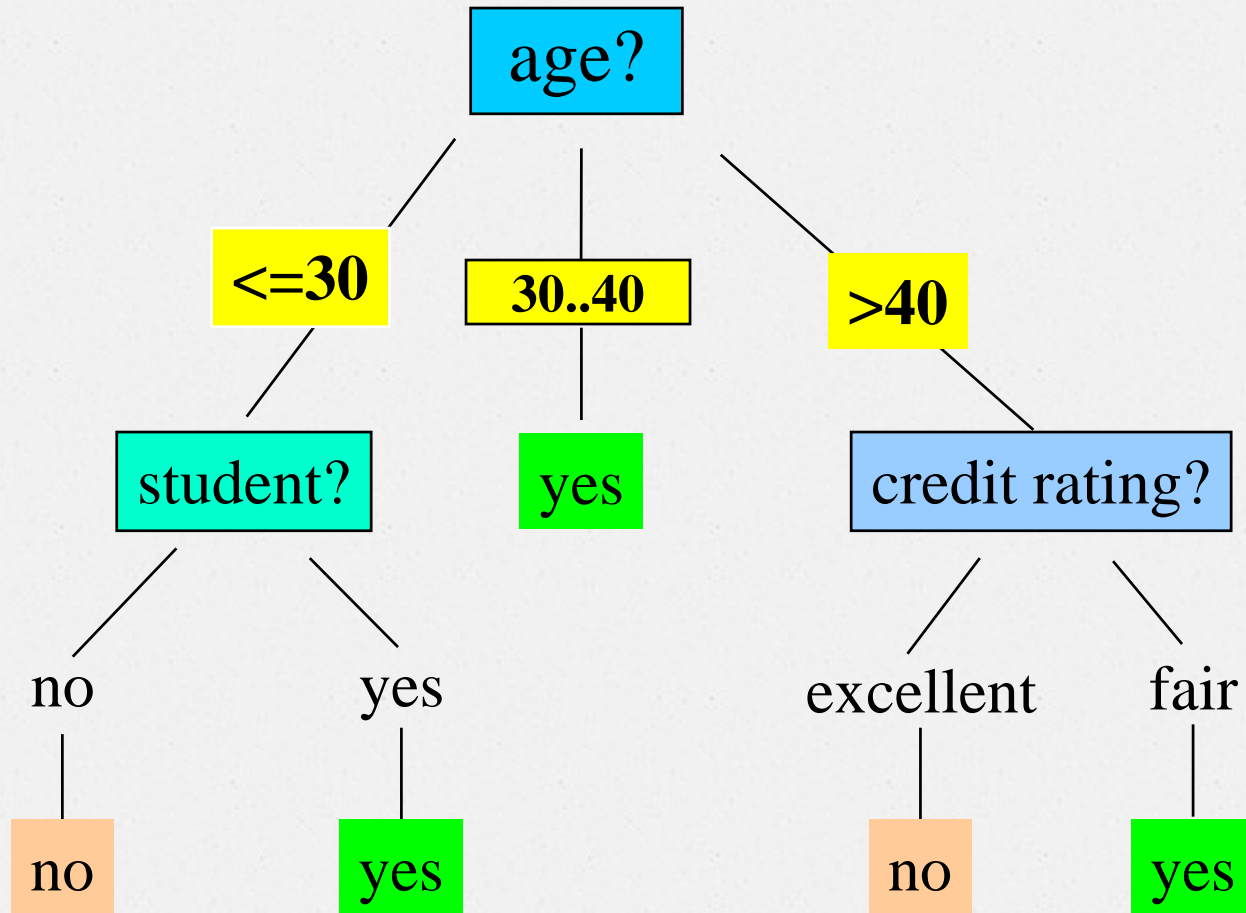
W_m مشخصه جداکننده

مثال خرید کامپیوتر توسط مشتری

o مجموعه داده های آموزشی

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no

نحوه نمایش درخت تصمیم



کلاس بندی درختها

○ فرایندی دو مرحله ای است

○ ساخت مدل :

- تحلیل یک مجموعه آموزشی که مجموعه ای از نمونه های پایگاه است و مشخص کردن برچسب کلاس های مربوط به این نمونه ها .
- هر کلاس با یک صفت که به آن صفت برچسب کلاس می گوئیم مشخص می شود .
- مجموعه آموزشی به صورت تصادفی از پایگاه انتخاب می شود .
- به این مرحله ، مرحله یادگیری نیز می گویند .

○ استفاده از مدل :

- از طریق یک تابع $y=f(X)$ برچسب کلاس هر نمونه X از پایگاه پیش بینی می شود .
- این تابع به صورت قواعد کلاسه بندی ، درخت های تصمیم گیری یا فرمول های ریاضی است .

اهداف اصلی دسته بندی کننده های درخت تصمیم

- (۱) داده های ورودی را تا حد ممکن درست دسته بندی کنند.
- (۲) دانش یادگیری شده از داده های آموزشی را به گونه ای عام سازی کنند که نمونه های نادیده بتوانند با دقت هرچه بیشتر دسته بندی شوند.
- (۳) بروزرسانی آسان درخت، در صورت اضافه شدن داده های آموزشی جدید، بتوان به راحتی درخت تصمیم گیری را گسترش داد (دارای خاصیت افزایشی باشند).
- (۴) داشتن یک ساختار تا سرحد امکان ساده.

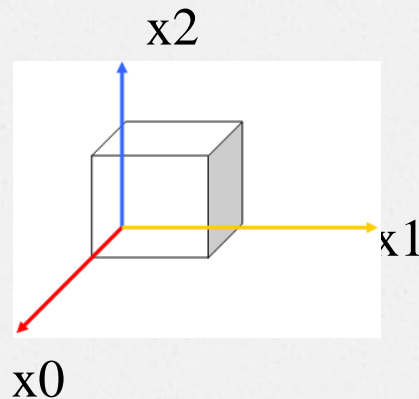
طبقه بندی بر اساس ویژگی

در درخت تصمیم (ID3) از یک مقدار آماری به نام بهره اطلاعات **Information Gain** استفاده می شود تا اینکه مشخص کنیم که یک ویژگی تا چه مقدار قادر است مثالهای آموزشی را بر حسب دسته بندی آنها جدا کند.

مثال

- برای یادگیری نحوه دسته بندی مثال ساده ای را بررسی می کنیم.
- در مثال ذیل کدام ویژگی باید در ریشه درخت قرار گیرد؟

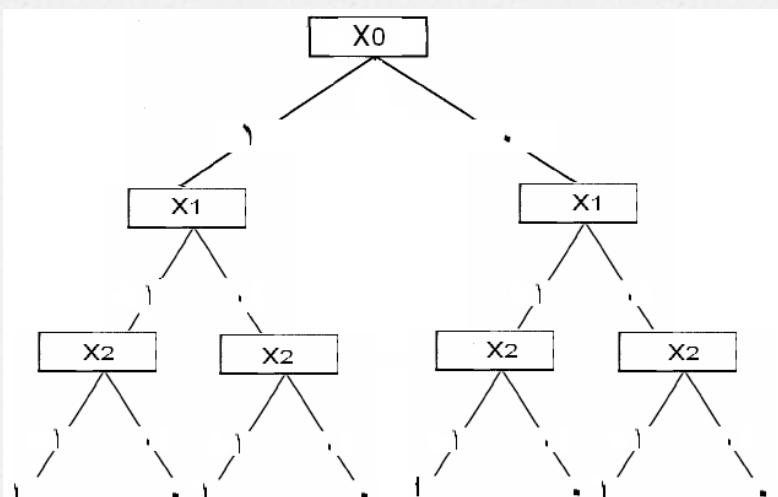
X_2	X_1	X_0	CLASS
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1



با توجه به صفحه عبوری از مکعب زیر x_0 یا x_2 می توانند ریشه باشند

حل مثال

درخت کامل ما به این صورت است:



این درخت کامل و پیچیده است و ضمناً خطا هم ندارد.

در بایاس کردن هم هدف

ساده کردن است که ممکن است ایجاد خطا کند.

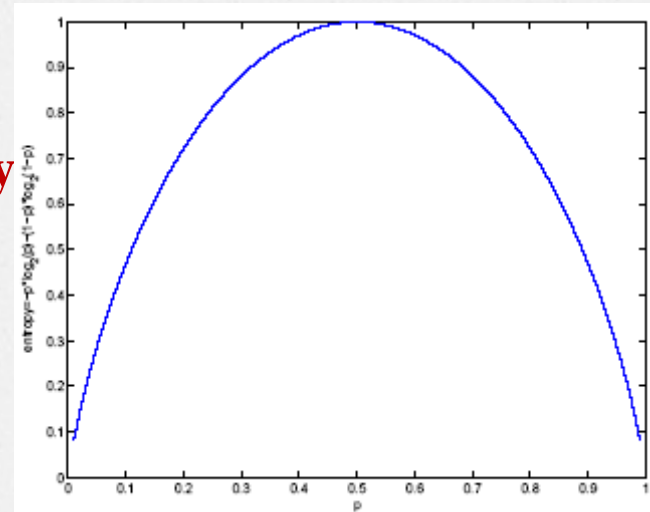
Classification Trees

- o For node m , N_m instances reach m , N_m^i belong to C_i

$$\hat{P}(C_i | \mathbf{x}, m) \equiv p_m^i = \frac{N_m^i}{N_m}$$

- o Node m is **pure** if p_m^i is 0 or 1
- o Measure of **impurity** is **entropy**

$$I_m = -\sum_{i=1}^K p_m^i \log_2 p_m^i$$



آنتروپی Entropy

• یک تابع برای اندازه گیری میزان خلوص را (بی نظمی یا **عدم خالص** بودن) **آنتروپی** می گویند. در نظریه اطلاعاتی، کمترین تعداد بیت‌های مورد نیاز برای رمز کردن کلاس بندی از یک نمونه را مشخص می کند.

• در هر **node**، خصوصیتی که بیشترین کاهش را در آنتروپی نمونه ها ایجاد می کند، انتخاب می شود

• وقتی $p^i = 1/k$ در این حالت حداکثر آنتروپی می تواند $\log_2 k$ باشد.

آنتروپی مثال

مثال: اگر مجموعه S شامل مثالهای مثبت و منفی از یک مفهوم هدف باشد آنتروپی S نسبت به این دسته بندی بولی بصورت زیر تعریف می شود.

$$Entropy(S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

که p_{\oplus} نسبت مثالهای مثبت به کل مثالها و p_{\ominus} نسبت مثالهای منفی به کل مثالها می باشد. همچنین $0 \log 0 = 0$ فرض میشود

مثال

• آنتروپی x_0 و x_1 و x_2 چقدر است؟

$$P_{\oplus} = \frac{2}{8}$$

$$P_{\ominus} = \frac{6}{8}$$

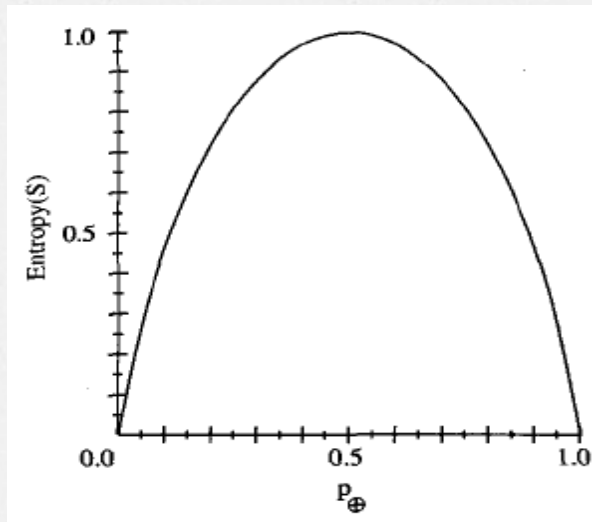
$$E = -\left(\frac{2}{8} \log_2 \frac{2}{8}\right) - \left(\frac{6}{8} \log_2 \frac{6}{8}\right)$$

• سؤال: اگر همه اعضاء S یکسان باشند آنتروپی چقدر است؟ **صفر**

سوال

0 اگر اعضای S نیمی مثبت و نیمی منفی باشد آنترופی چقدر است ؟

یک 0



Best Split

- o If node m is pure, generate a leaf and stop, otherwise split and continue recursively
- o Impurity after split: N_{mj} of N_m take branch j . N_{mj}^i belong to C_i

$$\hat{P}(C_i | \mathbf{x}, m, j) \equiv p_{mj}^i = \frac{N_{mj}^i}{N_{mj}}$$

$$I'_m = - \sum_{j=1}^n \frac{N_{mj}}{N_m} \sum_{i=1}^K p_{mj}^i \log_2 p_{mj}^i$$

- o Find the variable and split that min impurity (among all variables -- and split positions for numeric variables)

Split (Ex.)

GenerateTree(\mathcal{X})

If NodeEntropy(\mathcal{X}) < θ_I /* eq. 9.3

 Create leaf labelled by majority class in \mathcal{X}

 Return

$i \leftarrow$ SplitAttribute(\mathcal{X})

For each branch of \mathbf{x}_i

 Find \mathcal{X}_i falling in branch

 GenerateTree(\mathcal{X}_i)

SplitAttribute(\mathcal{X})

 MinEnt \leftarrow MAX

 For all attributes $i = 1, \dots, d$

 If \mathbf{x}_i is discrete with n values

 Split \mathcal{X} into $\mathcal{X}_1, \dots, \mathcal{X}_n$ by \mathbf{x}_i

$e \leftarrow$ SplitEntropy($\mathcal{X}_1, \dots, \mathcal{X}_n$) /* eq. 9.8 */

 If $e < \text{MinEnt}$ MinEnt \leftarrow e ; bestf \leftarrow i

 Else /* \mathbf{x}_i is numeric */

 For all possible splits

 Split \mathcal{X} into $\mathcal{X}_1, \mathcal{X}_2$ on \mathbf{x}_i

$e \leftarrow$ SplitEntropy($\mathcal{X}_1, \mathcal{X}_2$)

 If $e < \text{MinEnt}$ MinEnt \leftarrow e ; bestf \leftarrow i

 Return bestf

آنتروپی برای دسته بندی های غیر بولی

• اگر ویژگی هدف دارای C مقدار مختلف باشد آنتروپی S نسبت به این دسته بندی C گانه بصورت زیر تعریف میشود:

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

• که در آن p_i نسبتی از S است که به دسته i تعلق دارند. توجه شود که \log همچنان در مبنای ۲ گرفته میشود. در این حالت حداکثر آنتروپی میتواند $\log_2 C$ باشد.

بهره اطلاعات (Information Gain)

- بهره اطلاعات یک ویژگی عبارت است از مقدار کاهش آنتروپی که بواسطه جداسازی مثالها از طریق این ویژگی حاصل میشود.
- عبارت دیگر بهره اطلاعات $Gain(S, A)$ برای یک ویژگی نظیر A نسبت به مجموعه مثالهای S بصورت زیر تعریف میشود:

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

- که در آن $Values(A)$ مجموعه همه مقدار ویژگی های A بوده و S_v زیرمجموعه ای از S است که برای آن A دارای مقدار v است.
- در تعریف فوق عبارت اول مقدار آنتروپی داده ها و عبارت دوم مقدار آنتروپی مورد انتظار بعد از جداسازی داده هاست.

مثال

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
Day1	Sunny	Hot	High	Weak	No
Day2	Sunny	Hot	High	Strong	No
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	No
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	No
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	No

با استفاده از مثالهای
آموزشی زیر درختی
بسازید که لذت بخش
بودن بازی در
روزهایی با ویژگیهای
مختلف را تعیین کند.

$$\begin{aligned} \text{Entropy}([9+, 5-]) &= -(9/14) \log_2(9/14) - (5/14) \log_2(5/14) \\ &= 0.940 \end{aligned}$$

الگوریتم های یادگیری درخت تصمیم

○ اغلب الگوریتم های یادگیری درخت تصمیم بر پایه یک عمل جستجوی حریصانه (greedy) بالا به پائین (top-down) در فضای درختهای موجود عمل میکنند.

○ این الگوریتم پایه، **Concept Learning System (CLS)** نامیده می شود که در سال ۱۹۵۰ معرفی شده است.

○ این الگوریتم توسط Ross Quinlan در سال ۱۹۸۶ بصورت کاملتری تحت عنوان **Iterative Dichotomiser 3 (ID3)** (تکرار دوتایی) مطرح گردید.

○ بعدها الگوریتم کاملتر دیگری تحت عنوان **C4.5** ارائه گردید که برخی نقائص ID3 را برطرف میکند.

الگوریتم پایه ID3

این ایده توسط Ocuum's Razor بنا نهاده شده است. او می گوید:

“ دنیا ذاتا ساده است ”

بنابراین از کوچکترین درخت تصمیم که با داده سازگار باشد انتظار می رود که مثالهای نادیده را به درستی دسته بندی کند.

در این الگوریتم درخت تصمیم از بالا به پایین ساخته میشود. این الگوریتم با این سوال شروع میشود: کدام ویژگی باید در ریشه درخت مورد آزمایش قرار گیرد؟

برای یافتن جواب از یک آزمون آماری استفاده میشود تا مشخص گردد هر کدام تا چه حد قادر است به تنهایی مثالهای آزمایشی را دسته بندی کند. (که در اینجا از آنروپی استفاده می کنیم)

ID3 (ادامه...)

○ با انتخاب این ویژگی برای هر یک از مقادیر ممکن آن یک شاخه ایجاد شده و مثالهای آموزشی بر اساس ویژگی هر شاخه مرتب میشوند. بدنبال آن عملیات فوق برای مثالهای قرار گرفته در هر شاخه **تکرار** میشود تا بهترین ویژگی برای گره بعدی انتخاب شود.

○ این الگوریتم یک جستجوی حریصانه است که در آن انتخاب های قبلی هرگز مورد بازبینی قرار نمیگیرند.

○ این الگوریتم هر شاخه از درخت را آنقدر به عمق می برد که بتواند بطور کامل مثالهای آموزشی را دسته بندی کند این عمل میتواند منجر به **overfitting** شود.

Questions about the user/course under consideration. Ex:

- o **You:** Is the course under consideration in Systems?
- o **Me:** Yes
- o **You:** Has this student taken any other Systems courses?
- o **Me:** Yes
- o **You:** Has this student like most previous Systems courses?
- o **Me:** No
- o **You:** I predict this student will not like this course.

Overfitting

○ برای فرضیه ای مثل h متعلق به فضای فرضیه H دو نوع خطا تعریف میشود:

○ خطا روی داده های آموزشی $\text{error}_{\text{train}}(h)$

○ خطا روی کل داده های ممکن $\text{error}_D(h) \in D$

○ میگوئیم برای فرضیه $h \in H$ روی داده های آموزشی $\text{error}_{\text{train}}(h) < \text{error}_{\text{train}}(h_0)$ وجود Overfitting رخ میدهد اگر فرضیه ای مثل $h_0 \in H$ وجود داشته باشد که:

○ $\text{error}_{\text{train}}(h) < \text{error}_{\text{train}}(h_0)$

○ and

○ $\text{error}_D(h) > \text{error}_D(h_0)$

دلایل بروز Overfitting

○ الگوریتم ID3 هر شاخه از درخت را آنقدر به عمق میبرد که بتواند بطور کامل مثالهای آموزشی را دسته بندی کند. این امر میتواند منجر به Overfitting شود. دلایل بروز overfitting عبارتند از:

○ وجود نویز در داده های آموزشی

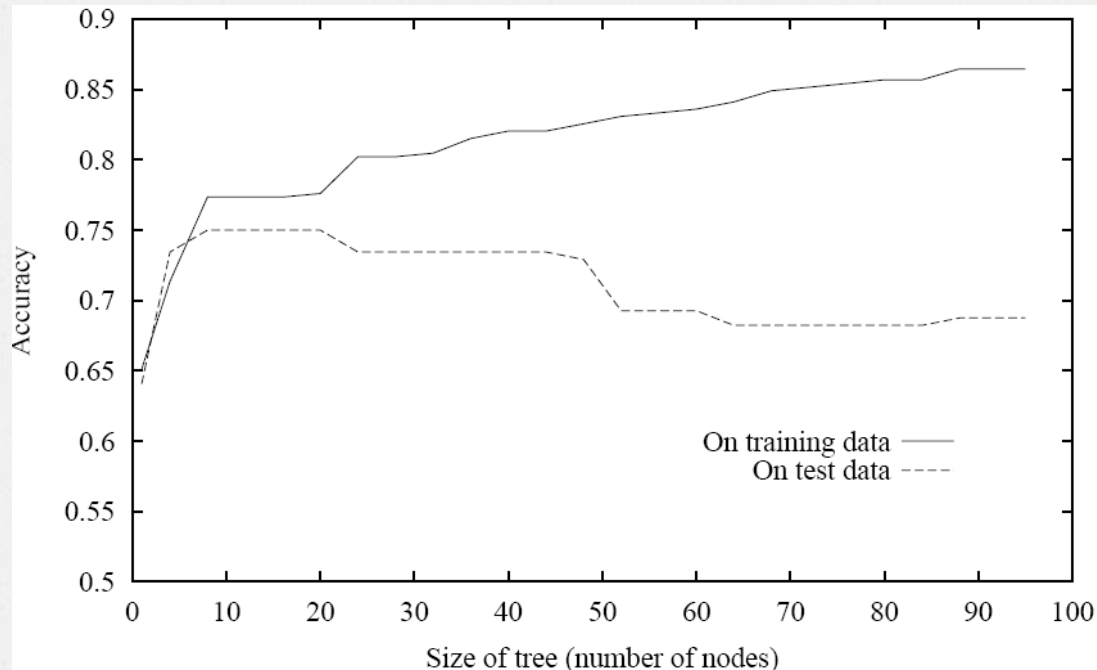
○ تعداد کم مثالهای آموزشی.

○ مثال: اگر فقط دو بار پرتاب سکه داشته باشیم و هر دو بار شیر آمده باشد چه نتیجه ای در مورد این آزمایش میتوان گرفت؟

پرهیز از Overfitting

1. جلوگیری از رشد درخت قبل از رسیدن به مرحله ای که بطور کامل داده های آموزشی را دسته بندی نماید.
2. اجازه به رشد کامل درخت و سپس حرس کردن شاخه هائی که مفید نیستند. (Post Pruning)
- 0 در عمل روش دوم بیشتر استفاده شده است زیرا تخمین اندازه صحیح درخت کار ساده ای نیست.

Sample of Overfitting on Trained & Test Data



Overfitting یک پدیده عمومی است

پدیده **overfitting** منحصر به درخت های تصمیم نیست و سایر روشهای یادگیری ماشینی نیز با آن مواجه هستند. این پدیده غالباً وقتی اتفاق می افتد که:

- o the hypothesis space is very large
- o the hypothesis search is not biased toward simple models
- o there is little training data
- o there is a lot of noise in the training data

در عمل با دیدن شرایط زیر میتوانیم بگوئیم که **overfitting** رخ داده است:

- o اختلاف زیاد بین دقت دسته بندی داده های آموزشی و داده های تست
- o رسیدن به فرضیه و یا مدل های خیلی پیچیده (مثلاً رسیدن به یک درخت تصمیم خیلی بزرگ)

در نظر گرفتن ویژگی های با مقادیر پیوسته

- درخت یاد گرفته شده توسط ID3 محدود به توابع و ویژگی های با مقدار گسسته است.
- برای اینکه این الگوریتم ویژگی های با مقدار پیوسته را نیز شامل شود، میتوان برای یک ویژگی پیوسته مثل A یک ویژگی بولی مثل AC تعریف کرد که AC درست است اگر $A < C$ باشد و در غیر اینصورت نادرست است.
- C باید طوری انتخاب شود که بهره اطلاعات را حداکثر کند. اینکار میتواند با مرتب کردن مقادیر ویژگی A و انتخاب نقاطی که مقادیر مثالهای مجاور تغییر میکنند انجام شود. در چنین حالتی میانگین دو مثال مجاور میتواند بعنوان آستانه انتخاب شود.

Temperature:	40	48	60	72	80	90
PlayTennis:	No	No	Yes	Yes	Yes	No

$$(48 + 60)/2$$

$$(80 + 90)/2$$

سایر معیارهای انتخاب ویژگی برای درخت

○ اگر به مثال قبل یک ویژگی به نام **تاریخ اضافه شود**، این ویژگی به تنهایی قادر خواهد بود تا کلیه مثالهای آموزشی را دسته بندی کند. در واقع بعلت اینکه این ویژگی دارای **بهره اطلاعات زیادی است** بعنوان ریشه درخت انتخاب خواهد شد و درخت حاصله دارای **عمق** بسیار **کمی** خواهد بود.

○ با وجود اینکه این درخت **مثالهای آموزشی را بخوبی دسته بندی خواهد کرد** اما در مورد **مثالهای نادیده بسیار ضعیف** عمل خواهد نمود. زیرا این درخت در عمل **مثالهای آموزشی را حفظ کرده و قادر به تعمیم نیست**.

الگوریتم - C4.5

○ الگوریتم - C4.5 نسل بعدی الگوریتم ID3 است و از نوعی از قانون حرس استفاده می کند. همچنین قادر است **صفات گسسته**، **صفات فاقد مقدار** و **داده های نویزی** را استفاده کند.

○ این الگوریتم **بهترین صفت** را با استفاده از معیار بی نظمی (عدم خالص بودن) انتخاب میکند و به دلیل استفاده از عامل **GainRatio** قادر به **بکارگیری صفات با مقادیر بسیار زیاد** می باشد.

○ حتی اگر هیچ خطایی در داده های آموزشی وجود نداشته باشد حرس انجام می شود که باعث می شود درخت عام تر شده و کمتر به مجموعه آموزشی وابسته شود.

قاعده استخراج از درخت

○ توانایی تفسیر کردن را دارند. گره ها حاصل شرایطی هستند که قابل فهمند.

○ از آنجائیکه تمام این شرایط باید حصول به برگ را جبران کنند این مسیرها در کنار هم می توانند به عنوان مجموعه ای از قواعد **IF-THEN** که اساس قواعد نامیده می شود مطرح شوند.

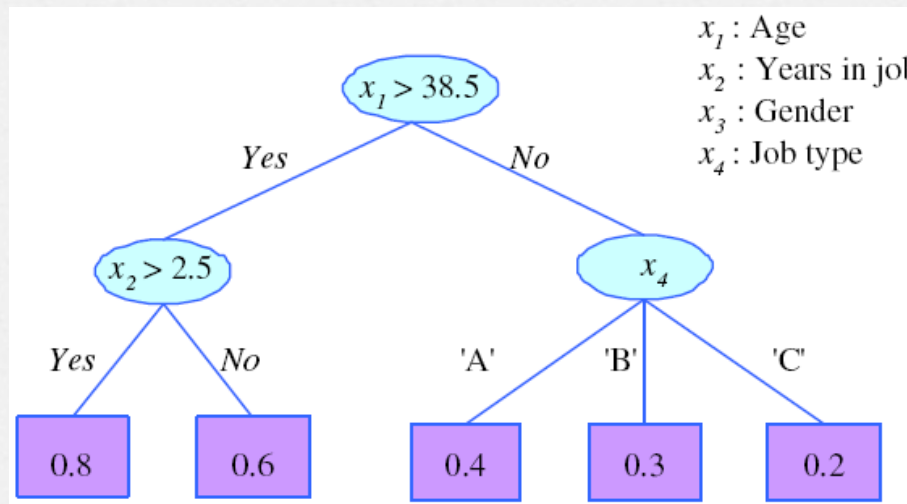
○ حرس کردن برای ساده سازی امکان پذیر است. در حرس کردن باید دقت داشته باشیم تا عناصر مهم را به اشتباه حذف نکنیم. برای مثال در عبارت اسلاید بعد می توانیم از سن صرف نظر کنیم

R3: IF (age ≤ 38.5) AND (job-type='A') THEN y = 0.4

R3' : IF (job-type='A') THEN y = 0.4

مثال: قاعدة استخراج از درخت

C4.5Rules
(Quinlan,
1993)



- R1: IF (age > 38.5) AND (years-in-job > 2.5) THEN $y = 0.8$
R2: IF (age > 38.5) AND (years-in-job \leq 2.5) THEN $y = 0.6$
R3: IF (age \leq 38.5) AND (job-type = 'A') THEN $y = 0.4$
R4: IF (age \leq 38.5) AND (job-type = 'B') THEN $y = 0.3$
R5: IF (age \leq 38.5) AND (job-type = 'C') THEN $y = 0.2$

معیار نسبت بهره یا Gain Ratio

- برای پرهیز از ویژگی هائی مثل تاریخ میتوان از معیار دیگری با نام نسبت بهره و یا **gain ratio** استفاده نمود که خاصیت آن حساسیت داشتن به این است که یک ویژگی با چه گستردگی و یکنواختی داده ها را جدا میکند.
- برای اینکار عبارتی بصورت زیر تعریف میشود:

$$SplitInformation(S, A) \equiv - \sum_{i=1}^c \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

- با استفاده از عبارت فوق نسبت بهره بصورت زیر تعریف میشود:

$$GainRatio(S, A) \equiv \frac{Gain(S, A)}{SplitInformation(S, A)}$$

معیار نسبت بهره یا Gain Ratio

○ SI(Split Info.) باعث میشود تا ویژگی هائی که مقادیر زیادی با توزیع یکنواخت دارند حذف گردند.

○ برای مثال یک ویژگی نظیر تاریخ برای تک تک مثالها توزیع یکسانی دارد از اینرو $SI = \log_2^n$ خواهد شد در حالیکه اگر یک ویژگی مثالها را به دو دسته تقسیم کند $SI=1$ خواهد شد.

○ یک مشکل عملی استفاده از معیار نسبت بهره این است که ممکن است مخرج این عبارت صفر و یا خیلی کوچک شود. در این حالت از روشهای هیوریستیک استفاده میشود.

ویژگی هائی با هزینه متفاوت

در بررسی پرونده یک بیمار ممکن است هزینه تست کردن برخی ویژگی ها بسیار بالا باشد و یا اینکه علیرغم موثر بودن ویژگی تست آن خطرناک باشد.

در این حالت باید درخت را طوری بایاس کرد که ویژگی های با هزینه کم را ترجیح دهد. برای مثال ممکن است بهره را بر هزینه تقسیم نمود.

مثالهایی با ویژگی های نامعلوم

در برخی از کاربردها نظیر مدارک پزشکی جمع آوری شده در بیمارستانهای مختلف ممکن است مقدار برخی از ویژگی ها درست ثبت نشده باشد. در این صورت یک انتخاب میتواند این باشد که به آن مقدار متداولترین مقدار مثالها در گروه n نسبت داده شود.

درخت تصمیم - ابزار دسته بندی

- معمولا بر مبنای کاهش آنتروپی عمل می کند.
- در هر ند، خصوصیتی که بیشترین کاهش را در آنتروپی نمونه ها ایجاد می کند، انتخاب می شود:

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

- نمونه ها بر حسب مقدار خصوصیت انتخاب شده در node جاری، بین زیر شاخه های آن تقسیم می شوند.
- هر node که تمام نمونه های آن از یک کلاس باشند، یک برگ است.

درخت تصمیم - یک مثال

مجموعه نمونه های آموزشی:

Id	Age	Education	Gender	Hours Per Week	Income	Buys
1	21-30	BS	Male	30-40	<50K	Yes
2	30-40	BS	Female	30-40	<50K	No
3	30-40	MS	Female	40-50	>50K	Yes
4	21-30	BS	Male	40-50	<50K	Yes
5	21-30	MS	Male	30-40	<50K	No
6	21-30	BS	Female	30-40	>50K	No

درخت تصمیم - یک مثال

- Entropy (S) = $-(3/6) \log(3/6) - (3/6) \log(3/6) \rightarrow$
Entropy (S) = 1
- Gain (S, Age) = $1 - ((4/6) + (2/6)) = 0$
- Gain (S, Education) = $1 - 1 = 0$
- Gain (S, Gender) = $1 - (-(2/3)\log(1/3) - (1/3)\log(1/3)) = 0.08$
- Gain (S, Hours) = $1 - 0.6 = 0.4$
- Gain (S, Income) = $1 - 1 = 0$
- Gain (S, Hours) = max Gain \rightarrow Root Node = Hours

Pseudocode

- o ID3 (Examples, Target_Attribute, Attributes)
- o Create a root node for the tree
- o If all examples are positive, Return the single-node tree Root, with label = +.
- o If all examples are negative, Return the single-node tree Root, with label = -.
- o If number of predicting attributes is empty, then Return the single node tree Root,
with label = most common value of the target attribute in the examples.
- o Otherwise Begin
 - o $A \leftarrow$ The Attribute that best classifies examples.
 - o Decision Tree attribute for Root = A.
 - o For each possible value, v_i , of A,
 - o Add a new tree branch below Root, corresponding to the test $A = v_i$.
 - o Let $\text{Examples}(v_i)$ be the subset of examples that have the value v_i for A
 - o If $\text{Examples}(v_i)$ is empty
 - o Then below this new branch add a leaf node with label = most common target value in the examples
 - o Else below this new branch add the subtree ID3 ($\text{Examples}(v_i)$, Target_Attribute, Attributes - {A})

کاربردها

- یکی از روش های کارآمد و با کاربرد گسترده کلاسه بندی است. (classification)
- درخت تصمیم در مسایلی کاربرد دارد که بتوان آنها را بصورتی مطرح نمود که پاسخ واحدی بصورت نام یک دسته یا کلاس ارائه دهند.
- برای مثال میتوان درخت تصمیمی ساخت که به این سوال پاسخ دهد: بیماری مریض کدام است؟ و یا درختی ساخت که به این سوال پاسخ دهد: آیا مریض به سرطان مبتلاست؟
- برای مسائلی مناسب است که مثالهای آموزشی بصورت زوج (مقدار-ویژگی) مشخص شده باشند.
- مثال: صفت مقدار: {گرم، معتدل، خنک} Temperature تابع هدف دارای خروجی با مقادیر گسسته باشد. مثلاً هر مثال با بله و خیر تعیین شود.

مزایای Decision Tree

1. قوانین تولید شده و به کار گرفته شده قابل استخراج و قابل فهم می باشند.
2. درخت تصمیم، توانایی کار با داده های پیوسته و گسسته را دارد. (روش های دیگر فقط توان کار با یک نوع را دارند. مثلاً شبکه های عصبی فقط توان کار با داده های پیوسته و قوانین رابطه با داده های گسسته)
3. درخت تصمیم از نواحی تصمیم گیری ساده استفاده می کند.
4. مقایسه های غیر ضروری در این ساختار حذف می شود. (حرس کردن)
5. تایید یک مدل در درخت های تصمیم با استفاده از تست های آماری امکان پذیر است. (قابلیت اطمینان مدل را می توان نشان داد)
6. روابط غیر منتظره یا نامعلوم را می یابند.
7. درخت های تصمیم قادر به سازگار کردن داده های فاقد مقدار می باشند.

معایب Decision Tree

1. در موارد با تعداد دسته های زیاد و نمونه آموزشی کم، احتمال خطا بالاست.
2. تولید درخت تصمیم گیری، هزینه محاسباتی بالا دارد.
3. هرس کردن درخت هزینه بالایی دارد.
4. در مسائلی که دسته های ورودی با نواحی مکعبی به خوبی جدا نشوند و دسته ها همپوشانی داشته باشند، خوب عمل نمی کنند.
5. در صورت همپوشانی گره ها تعداد گره های پایانی زیاد می شود.
6. طراحی درخت تصمیم گیری بهینه، دشوار است. کارایی یک درخت دسته بندی کننده به چگونگی طراحی خوب آن بستگی دارد.
7. احتمال تولید روابط نادرست وجود دارد.
8. بازنمایی درخت تصمیم دشوار است.
9. وقتی تعداد دسته ها زیاد است، می تواند باعث شود که تعداد گره های پایانی بیشتر از تعداد دسته های واقعی بوده و بنابراین زمان جستجو و فضای حافظه را افزایش می دهد.

موفق باشید