



یادگیری ماشین Machine Learning

یادگیری درخت تصمیم

Decision Tree Learning

ارائه دهنده: دکتر سیدین
دانشکده مهندسی برق - دانشگاه صنعتی امیرکبیر

نیمسال دوم ۹۶-۹۷

یادگیری درخت تصمیم - مقدمه

- جزو مشهورترین الگوریتم‌های استنتاج استقرایی
- برای تقریب توابع هدف با مقادیر گسته
- نمایش تابع یاد گرفته شده توسط یک درخت مقاوم به داده نویزی
- توانایی یادگیری بیان‌های فصلی
- جستجوی یک فضای فرضیه رسا و کامل
- عدم وجود مشکلات فضاهای فرضیه محدود
- امکان نمایش مجدد بصورت مجموعه قوانین if-then

نمایش درخت تصمیم

□ پادگیری

□ چگونگی طبقه‌بندی نمونه‌ها توسط درخت تصمیم:

■ مرتب کردن (sort) نمونه‌ها به پایین درخت از ریشه به سمت
تعدادی برگ

□ هر گره درخت: مشخص کننده یک ویژگی (attribute) از یک
نمونه

□ هر شاخه پایین رفته از آن گره: مطابق با یکی از مقادیر ممکن
این ویژگی

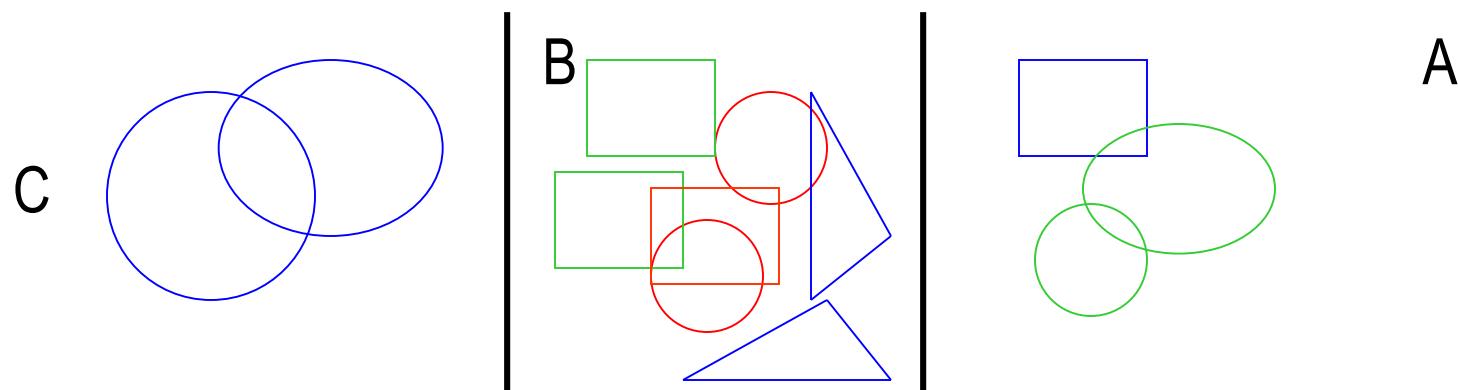
نمایش درخت تصمیم

طبقه‌بندی نمونه: □

- شروع از گره (node) ریشه درخت
- تست کردن ویژگی مشخص شده توسط این گره
- حرکت به سمت پایین درخت در آن شاخه مرتبط به مقدار ویژگی داده شده در مثال
- تکرار این پروسه برای زیردرخت منشعب شده در گره جدید

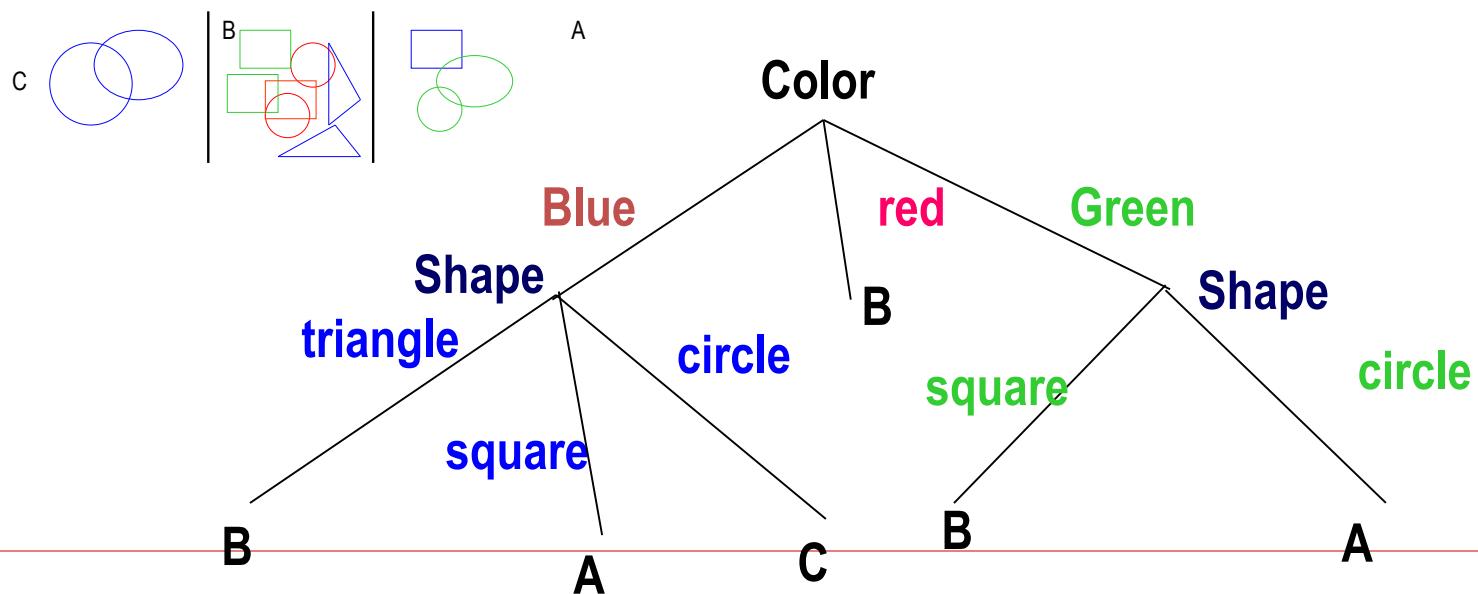
نمایش درخت تصمیم

- ❑ مثال: طبقه‌بندی نمونه‌هایی با بردارهای ویژگی:
- ❑ (`color= ;shape= ;label=`)



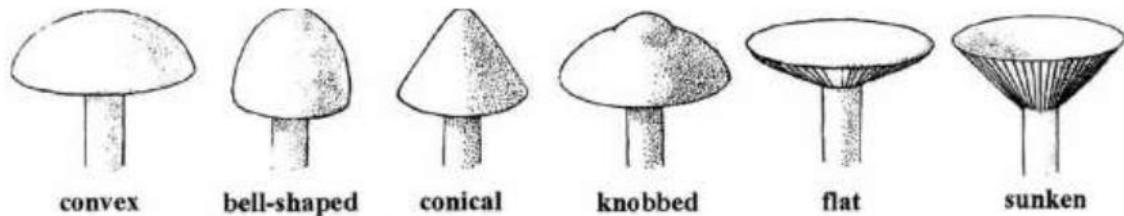
نمایش درخت تصمیم

- گره‌ها یا تست‌ها برای مقادیر ویژگیها
- یک شاخه برای هر مقدار ویژگی
- برگ‌ها: مشخص کننده برچسب‌ها

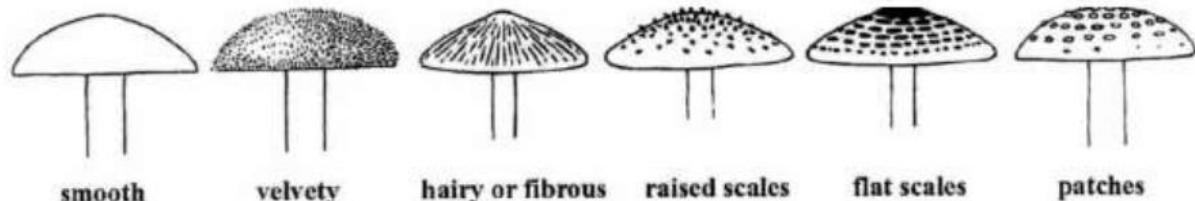


مثال: نمونه‌ای از کاربرد در طبقه‌بندی قارچ‌ها به سمی و خوراکی

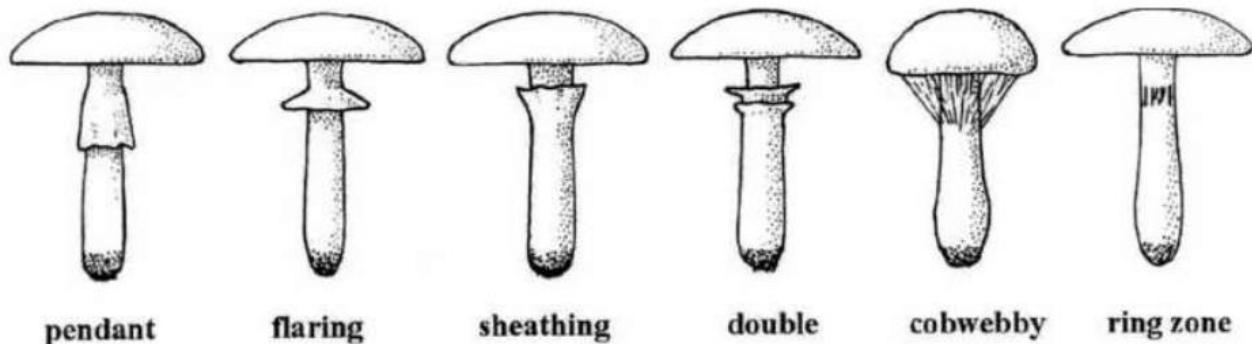
Mushroom cap shapes



Mushroom cap surfaces



Annular rings



مثال کاربرد در طبقه‌بندی قارچ‌ها به سمی و خوراکی براساس ویژگیهای مختلف

1. cap-shape: bell=b,conical=c,convex=x,flat=f,
knobbed=k,sunken=s
2. cap-surface: fibrous=f,grooves=g,scaly=y,smooth=s
3. cap-color: brown=n,buff=b,cinnamon=c,gray=g,green=r,
pink=p,purple=u,red=e,white=w,yellow=y
4. bruises?: bruises=t,no=f
5. odor: almond=a,anise=l,creosote=c,fishy=y,foul=f,
musty=m,none=n,pungent=p,spicy=s
6. gill-attachment: attached=a,descending=d,free=f,notched=n
7. ...

مثال کاربرد در طبقه‌بندی قارچ‌ها به سمی و خوراکی

۲ نمونه قارچ □

□ p: poisonous, e: edible

$x_1 = x, s, n, t, p, f, c, n, k, e, e, s, s, w, w, p, w, o, p, k, s, u$

$y_1 = p$

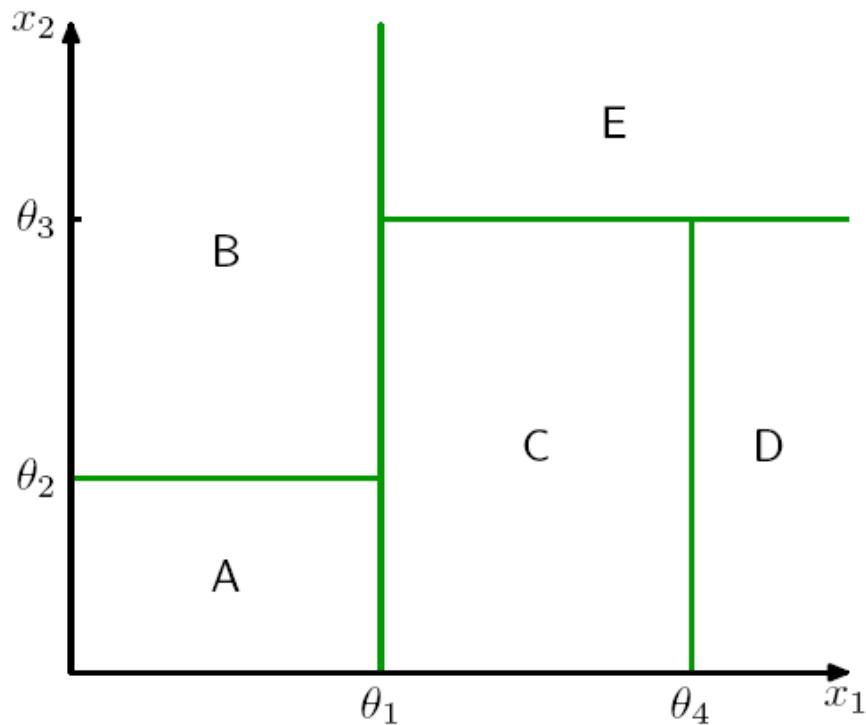
$x_2 = x, s, y, t, a, f, c, b, k, e, c, s, s, w, w, p, w, o, p, n, n, g$

$y_2 = e$

نمایش درخت تصمیم

مثال: پارسیشن بندی فضای ۲بعدی ورودی به نواحی با مرزهای موازی با محورها

Figure 14.5 Illustration of a two-dimensional input space that has been partitioned into five regions using axis-aligned boundaries.

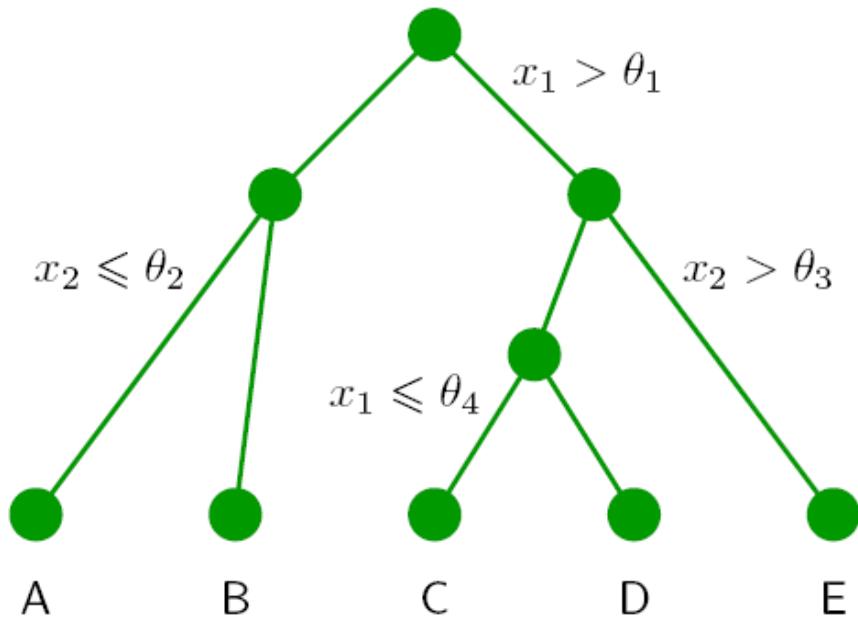


نمایش درخت تصمیم - مثال پارتیشن بندی فضای ۲ بعدی

ورودی

□ درخت تصمیم باینری: در هر گره به ۲ شاخه تقسیم می‌شود.

Figure 14.6 Binary tree corresponding to the partitioning of input space shown in Figure 14.5.



نمایش درخت تصمیم

مثال: طبقه‌بندی صبح‌های شنبه براساس اینکه آیا مناسب بازی تنیس هستند یا نه.

Day	Outlook	Temperature	Humidity	Wind	PlayTennis?
1	Sunny	Hot	High	Light	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Light	Yes
4	Rain	Mild	High	Light	Yes
5	Rain	Cool	Normal	Light	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Light	No
9	Sunny	Cool	Normal	Light	Yes
10	Rain	Mild	Normal	Light	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Light	Yes
14	Rain	Mild	High	Strong	No

نمایش درخت تصمیم

مثال

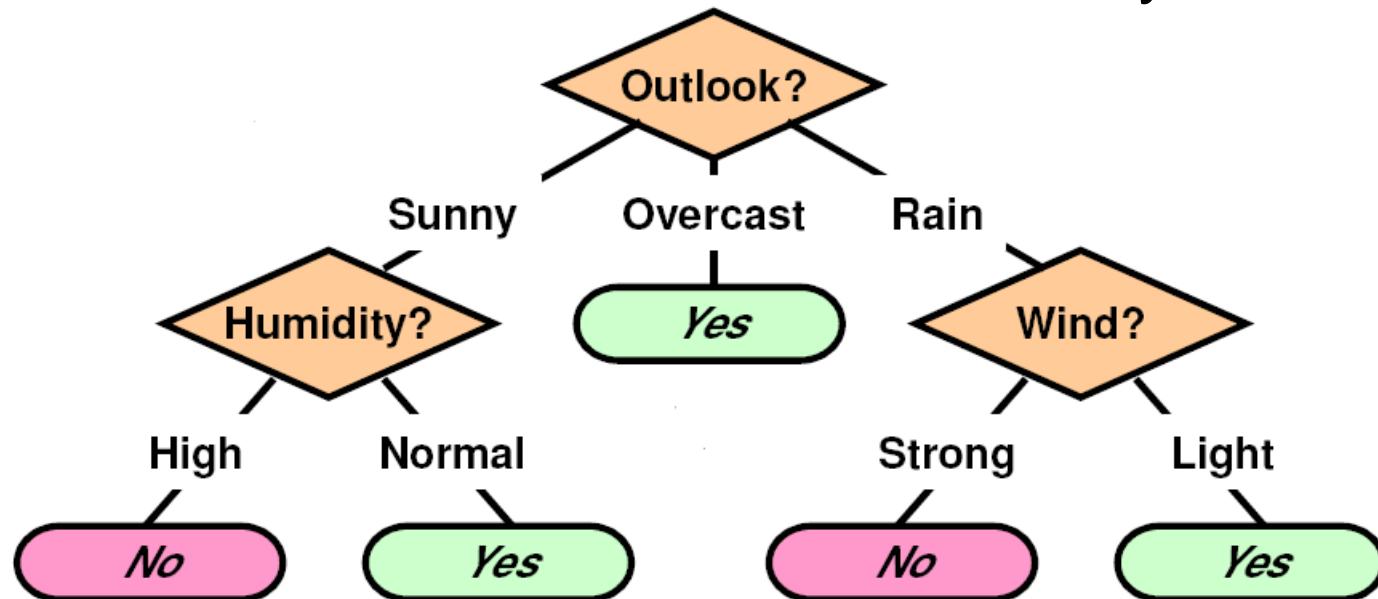


FIGURE 3.1

A decision tree for the concept *PlayTennis*. An example is classified by sorting it through the tree to the appropriate leaf node, then returning the classification associated with this leaf (in this case, *Yes* or *No*). This tree classifies Saturday mornings according to whether or not they are suitable for playing tennis.

نمایش درخت تصمیم

:PlayTennis

- <Outlook=sunny, Temperature=Hot, Humidity= High, Wind=Strong>
→ *PlayTennis=no*

در چپ‌ترین شاخه درخت تصمیم ← طبقه‌بندی بصورت نمونه منفی

نمایش درخت تصمیم

- در حالت کلی: درخت‌های تصمیم، یک رابطه فصلی (conjunction) از ترکیب عطفی (disjunction) مقادیر ویژگی نمونه‌ها را نمایش می‌دهند:
 - رابطه عطفی: هر مسیر از ریشه درخت به یک برگ
 - رابطه فصلی: خود درخت
- در مثال :*PlayTennis*

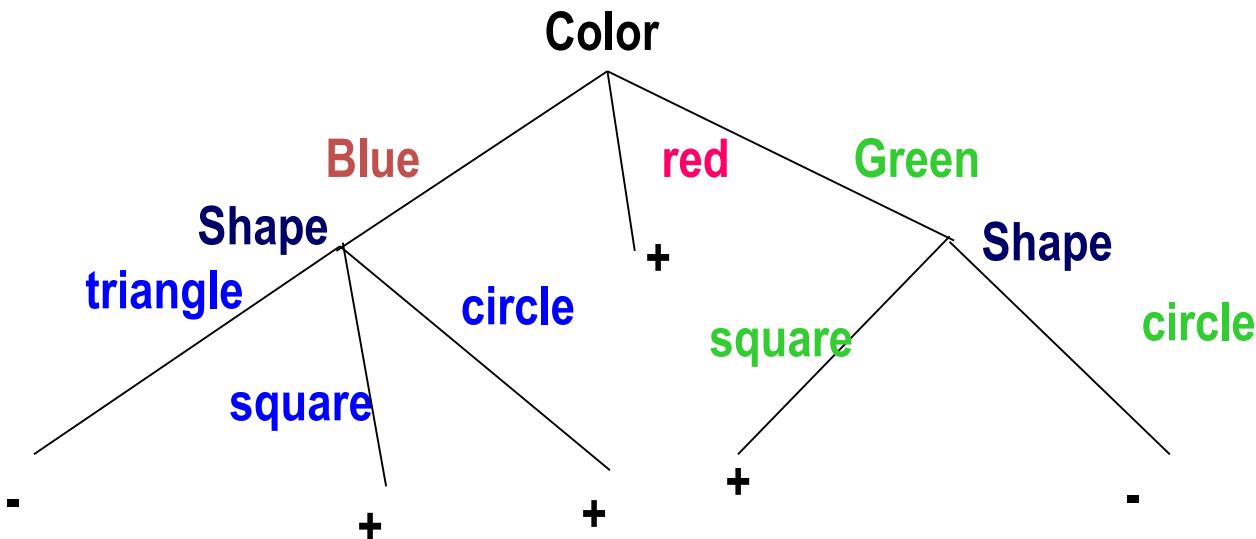
$(Outlook = Sunny \wedge Humidity = Normal)$

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

نمایش درخت تصمیم

□ نمونه دیگری از یک تابع بولی با تعریف قوانین به فرم فصلی:

- green \wedge square \rightarrow positive
- blue \wedge circle \rightarrow positive
- blue \wedge square \rightarrow positive
- The disjunction of these rules is equivalent to the Decision Tree



مسائل مناسب برای یادگیری درخت تصمیم

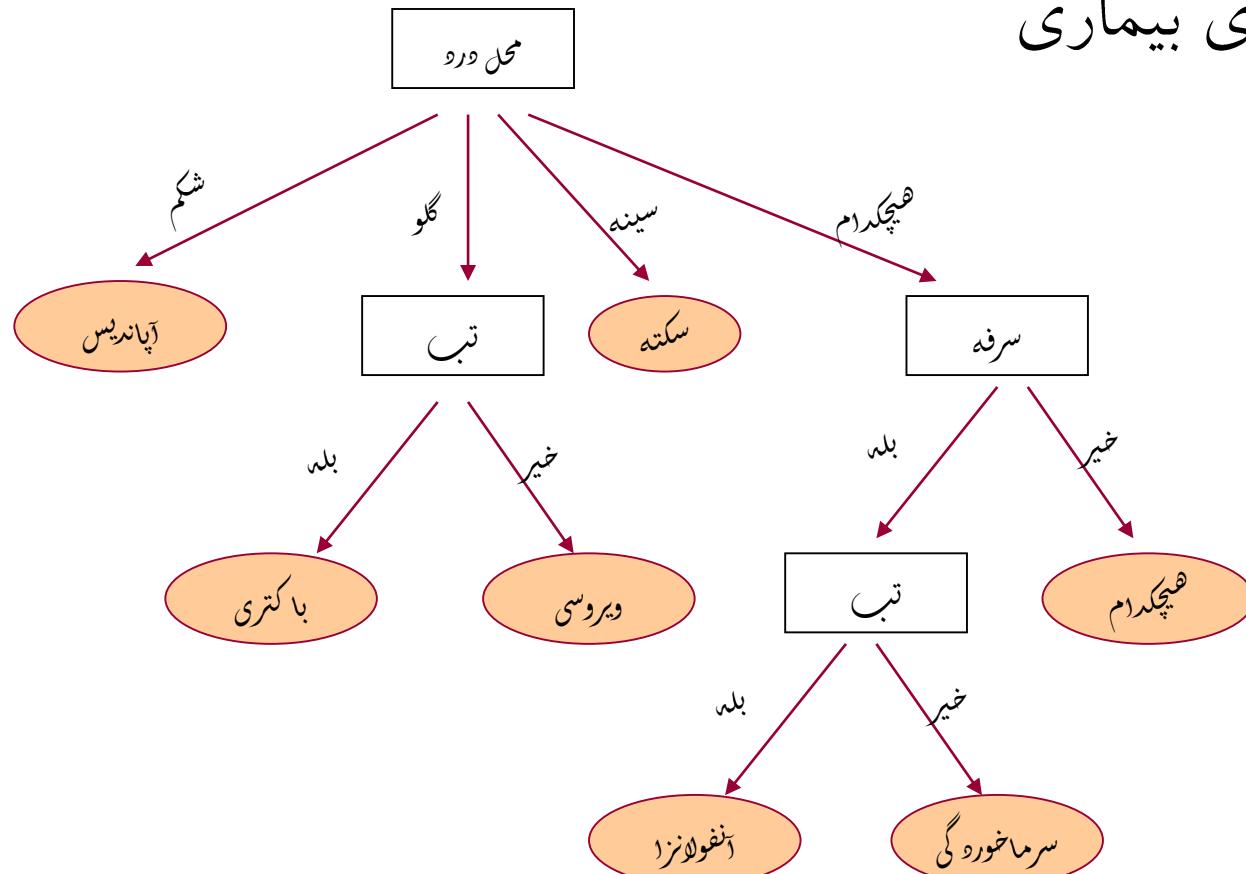
- نمونه‌ها توسط زوج‌های (ویژگی-مقدار) نمایش داده می‌شوند.
- ساده‌ترین موقعیت: هر ویژگی تعدادی مقدار کم از مقادیر ممکن (Hot,Mild,Cold) بگیرد. مثلا disjoint
- تابع هدف مقادیر خروجی گستته دارد.
- توصیف‌های فصلی (disjunctive) لازم است.
- داده‌های آموزشی می‌توانند خطأ داشته باشند. مقاومت به خطاهای:
- طبقه‌بندی مثال‌های آموزشی
- در مقادیر ویژگی توصیف کننده این مثال‌ها
- داده‌های آموزشی می‌توانند قادر برخی مقادیر ویژگی‌ها باشند.

مسائل مناسب برای یادگیری درخت تصمیم

- ❑ مسائل عملی متعدد منطبق با این خاصیت‌ها:
 - یاد گرفتن طبقه‌بندی بیماران براساس بیماری‌شان
 - کارکرد بد دستگاه‌ها براساس دلیل آنها
 - متقاضیان وام براساس درست‌نمایی تخلف در پرداخت

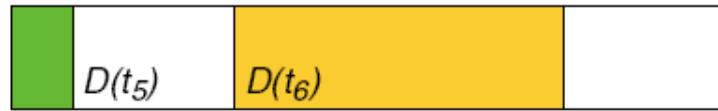
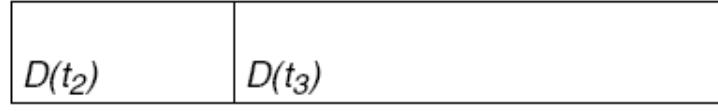
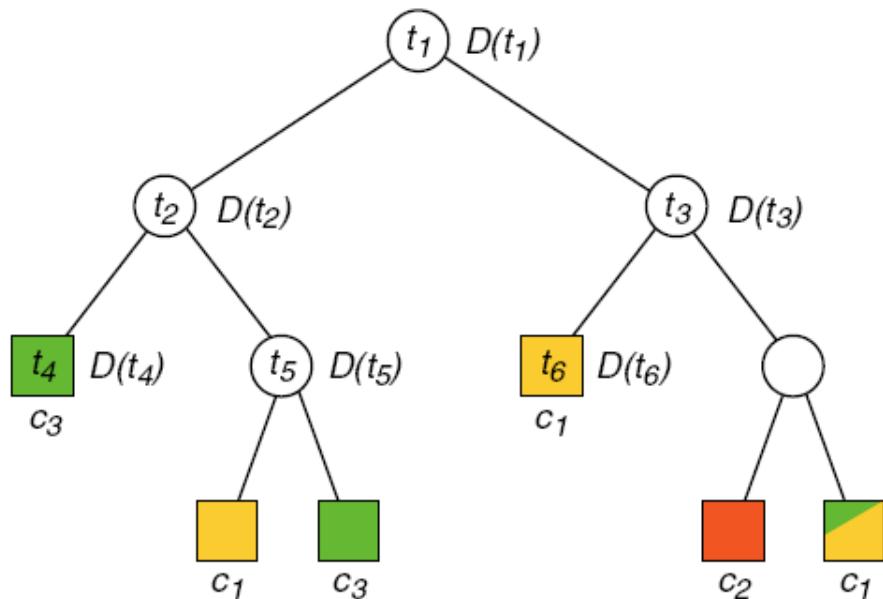
مسائل مناسب برای یادگیری درخت تصمیم

□ کاربرد طبقه‌بندی بیماری



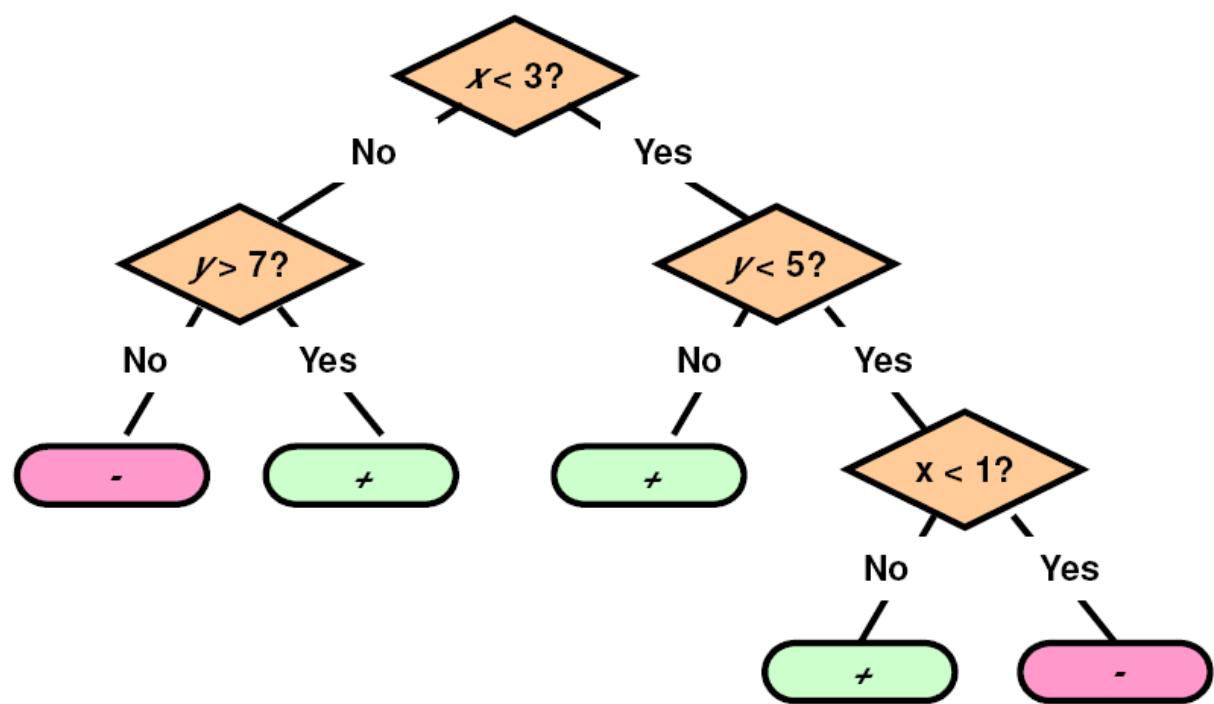
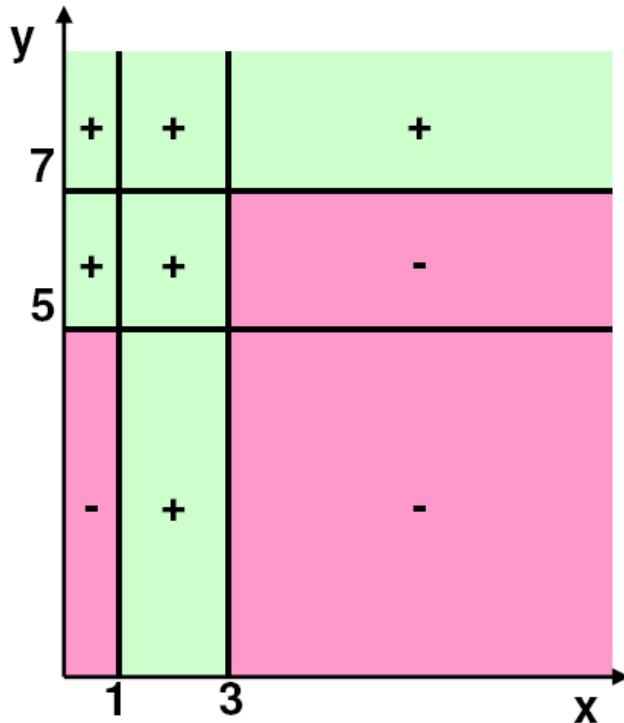
مسائل مناسب برای یادگیری درخت تصمیم

□ مثال چند مقدار برای تابع هدف



مسائل مناسب برای یادگیری درخت تصمیم

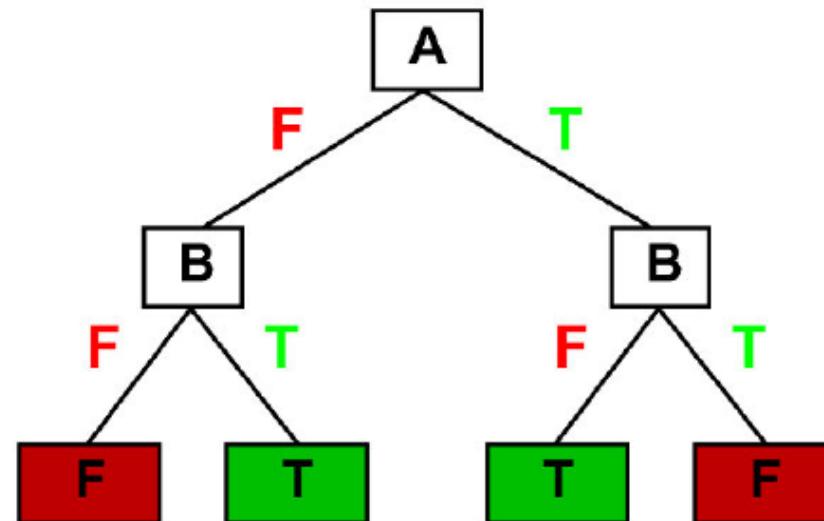
□ درخت تصمیم فضای نمونه‌ها را به مستطیل‌هایی موازی با محورها تقسیم می‌کند.



مسائل مناسب برای یادگیری درخت تصمیم

- درخت تصمیم می‌تواند هر تابعی از ویژگی‌های ورودی را نمایش دهد. مثلاً در توابع بولی:
- هر ردیف جدول درستی: مسیر برگ درخت

A	B	$A \text{ xor } B$
F	F	F
F	T	T
T	F	T
T	T	F



الگوریتم یادگیری درخت تصمیم

- بر پایه تغییرات یک الگوریتم مرکزی در فضای درخت‌های تصمیم ممکن:
 - جستجوی بالا به پایین top-down
 - الگوریتم حریصانه greedy
- یک الگوریتم: ID3
- نوع کامل‌تر ID3 : C4.5

الگوریتم ID3

- یادگیری درخت‌های تصمیم با ساختن آنها از بالا به پایین
براساس یک سوال اصلی:
 - کدام ویژگی باید در ریشه درخت تست شود؟
 - ارزیابی هر ویژگی نمونه براساس یک تست آماری و تعیین اینکه چقدر خوب می‌تواند به تنها یعنی مثال‌های آموزشی را طبقه‌بندی کند.
- یک جستجوی حریصانه که در آن الگوریتم هیچ وقت برای بررسی مجدد انتخاب‌های قبلی به عقب برنمی‌گردد.

الگوریتم ID3

- ۱- انتخاب بهترین ویژگی و قرار دادن در گره ریشه درخت
- ۲- ایجاد شاخه برای هر مقدار ممکن این ویژگی
- ۳- مرتب کردن مثال‌های آموزشی براساس ویژگی هر شاخه
- ۴- تکرار کل پروسه با مثال‌های قرار گرفته در هر شاخه و انتخاب بهترین ویژگی برای گره بعدی

خلاصه الگوریتم ID3 برای یادگیری توابع بولی

ID3(*Examples*, *Target_attribute*, *Attributes*)

Examples are the training examples. *Target_attribute* is the attribute whose value is to be predicted by the tree. *Attributes* is a list of other attributes that may be tested by the learned decision tree. Returns a decision tree that correctly classifies the given *Examples*.

- Create a *Root* node for the tree
- If all *Examples* are positive, Return the single-node tree *Root*, with label = +
- If all *Examples* are negative, Return the single-node tree *Root*, with label = -
- If *Attributes* is empty, Return the single-node tree *Root*, with label = most common value of *Target_attribute* in *Examples*
- Otherwise Begin
 - $A \leftarrow$ the attribute from *Attributes* that best* classifies *Examples*
 - The decision attribute for *Root* $\leftarrow A$
 - For each possible value, v_i , of *A*,
 - Add a new tree branch below *Root*, corresponding to the test $A = v_i$
 - Let $Examples_{v_i}$ be the subset of *Examples* that have value v_i for *A*
 - If $Examples_{v_i}$ is empty
 - Then below this new branch add a leaf node with label = most common value of *Target_attribute* in *Examples*
 - Else below this new branch add the subtree
$$ID3(Examples_{v_i}, Target_attribute, Attributes - \{A\})$$
- End
- Return *Root*

* The best attribute is the one with highest *information gain*, as defined in Equation (3.4).

کدام ویژگی بهترین طبقه‌بندی کننده است؟

- انتخاب مرکزی در ID3: اینکه کدام ویژگی باید در هر گره درخت تست شود.
- هدف ما: انتخاب مفیدترین ویژگی برای طبقه‌بندی مثال‌ها
- یک معیار اندازه‌گیری کمی: ویژگی آماری به نام **بهره اطلاعات** **information gain** یا
- اندازه‌گیری اینکه یک ویژگی داده شده چقدر خوب می‌تواند مثال‌های آموزشی را براساس طبقه‌بندی هدف آنها جداسازی کند.
- استفاده ID3 از information gain برای انتخاب بین ویژگی‌های کاندید در هر گام رشد درخت

آنتروپی Entropy

□ تعریف: ناچالصی (impurity) یک مجموعه دلخواه از مثال‌ها را مشخص می‌کند.

□ در حالت مفهوم هدف باینری با داشتن مجموعه S از مثال‌های مثبت و منفی:

$$\text{Entropy}(S) = -p_+ \log_2(p_+) - p_- \log_2(p_-)$$

□ مثال: S شامل ۱۴ مثال + و - [9+, 5-]

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

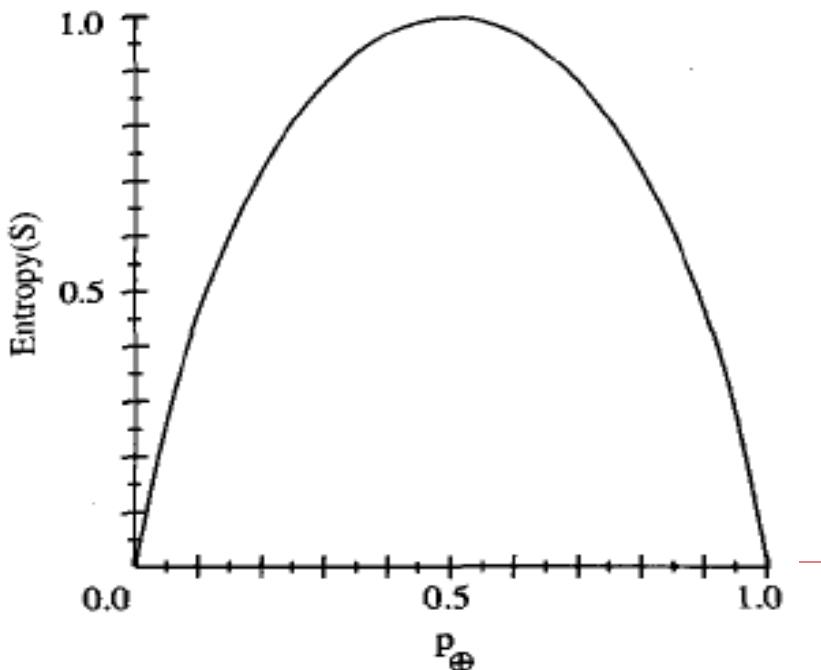
آنتروپی Entropy

اگر تمام اعضای S متعلق به ۱ کلاس باشند:

اگر تعداد مثال‌های + و - مساوی باشند:

اگر در مجموعه S ، تعداد مثال‌های + و - یکی نباشند:

$0 < \text{Entropy} < 1$



مفهوم آنتروپی

- شروع با بحث کدگذاری و مشخص کردن تعداد بیت مورد نیاز برای ارسال
- مثال: اگر یک میلیون بار یک سکه را بیندازیم و بخواهیم نتایج آن را برای فرد دیگری ارسال کنیم، به چه تعداد بیت نیاز داریم؟
 - ۲ حالت داریم:
 - سکه سالم
 - سکه ناسالم

مفهوم آنتروپی

۱- مثال سکه سالم:

تعداد بیت‌ها برای نمایش وضعیت هر پرتاب (H یا T): ۱ بیت

تعداد بیت‌ها برای ۱ میلیون پرتاب:

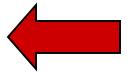
۲- مثال سکه ناسالم:

$P(H)=1/1000$, $P(T)=999/1000$
احتمال اینکه در ۱ میلیون بار شیر بیاید:

$$P(H) \times 10^6 = 1000$$

مفهوم آنتروپی

□ ۲- مثال سکه ناسالم:

- ❖ یک راه: ارسال شماره حالتها یی که شیر آمده، به جای ارسال وضعیت هر پرتاپ:
- ❖ شماره هر پرتاپ عددی بین ۱ تا 1000000 : قابل نمایش با 20 بیت
- ❖ در 1 امیلیون بار: 1000 بار شیر → 20000 بیت
- ❖ دنباله نمایش سکه‌های ناسالم، اطلاعات کمتری نسبت به سکه‌های سالم دارد.

- ❖ نیاز به بیت‌های کمتری برای انتقال

مفهوم آنتروپی

- می‌توانیم کدهای با طول متغیر تعریف کنیم:
- مثال: در فرستادن ۴ سمبول A و B و C و D
- 1) if $P(X=A)=P(X=B)=P(X=C)=P(X=D)=1/4$
 - We need 2 bits for 4 symbols:

0	0	A
0	1	B
1	0	C
1	1	D

مفهوم آنتروپی - کدهای با طول متغیر

- مثال: در فرستادن ۴ سمبول A و B و C و D
- 1) if $P(X=A)=1/2$, $P(X=B)=1/4$, $P(X=C)=1/8$,
 $P(X=D)=1/8$

❖ می‌توان تعداد بیت کمتری در کدگذاری داشت. مثلا:

A	0
B	10
C	110
D	111

(احتمال وقوع آن سمبول) \times (تعداد بیت هر سمبول) = متوسط تعداد بیت

$$\text{Avg. Bits} = 1 \times 1/2 + 2 \times 1/4 + 3 \times 1/8 + 3 \times 1/8 = 1.75 \text{ bits}$$

❖ اختصاص تعداد بیت بیشتر به سمبول‌های با احتمال وقوع کمتر

مفهوم آنتروپی - کدهای با طول متغیر

حالت کلی: فرض کنید X می‌تواند m مقدار V_1 تا V_m بگیرد. □

کمترین تعداد متوسط ممکن بیت در هر سمبول لازم برای ارسال رشته سمبول‌ها:

$$\overline{P(X=V_1) = p_1 \quad P(X=V_2) = p_2 \quad \dots \quad P(X=V_m) = p_m}$$

$$\begin{aligned} H(X) &= -p_1 \log_2 p_1 - p_2 \log_2 p_2 - \dots - p_m \log_2 p_m \\ &= -\sum_{j=1}^m p_j \log_2 p_j = Entropy \ (X) \end{aligned}$$

مفهوم انتروپی: مشخص کننده میزان ناچالصی

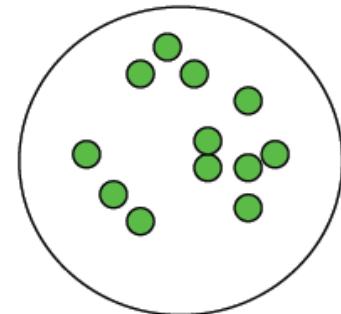
در حالت ۲ کلاسه:

- What is the entropy of a group in which all examples belong to the same class?

Entropy=0

not a good training set for learning

Minimum impurity

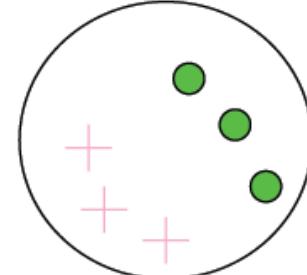


- What is the entropy of a group with 50% in either class?

Entropy=1

good training set for learning

Maximum impurity



مفهوم آنتروپی

X: High Entropy □
یک توزیع یکنواخت دارد.

- مقادیر نمونه برداری شده از آن می‌توانند هر جا باشند.
- میزان عدم قطعیت بیشتر

X: Low Entropy □
توزیع شامل قله‌ها و دره‌های است.

- مقادیر نمونه برداری شده از آن بیشتر قابل پیش‌بینی هستند.

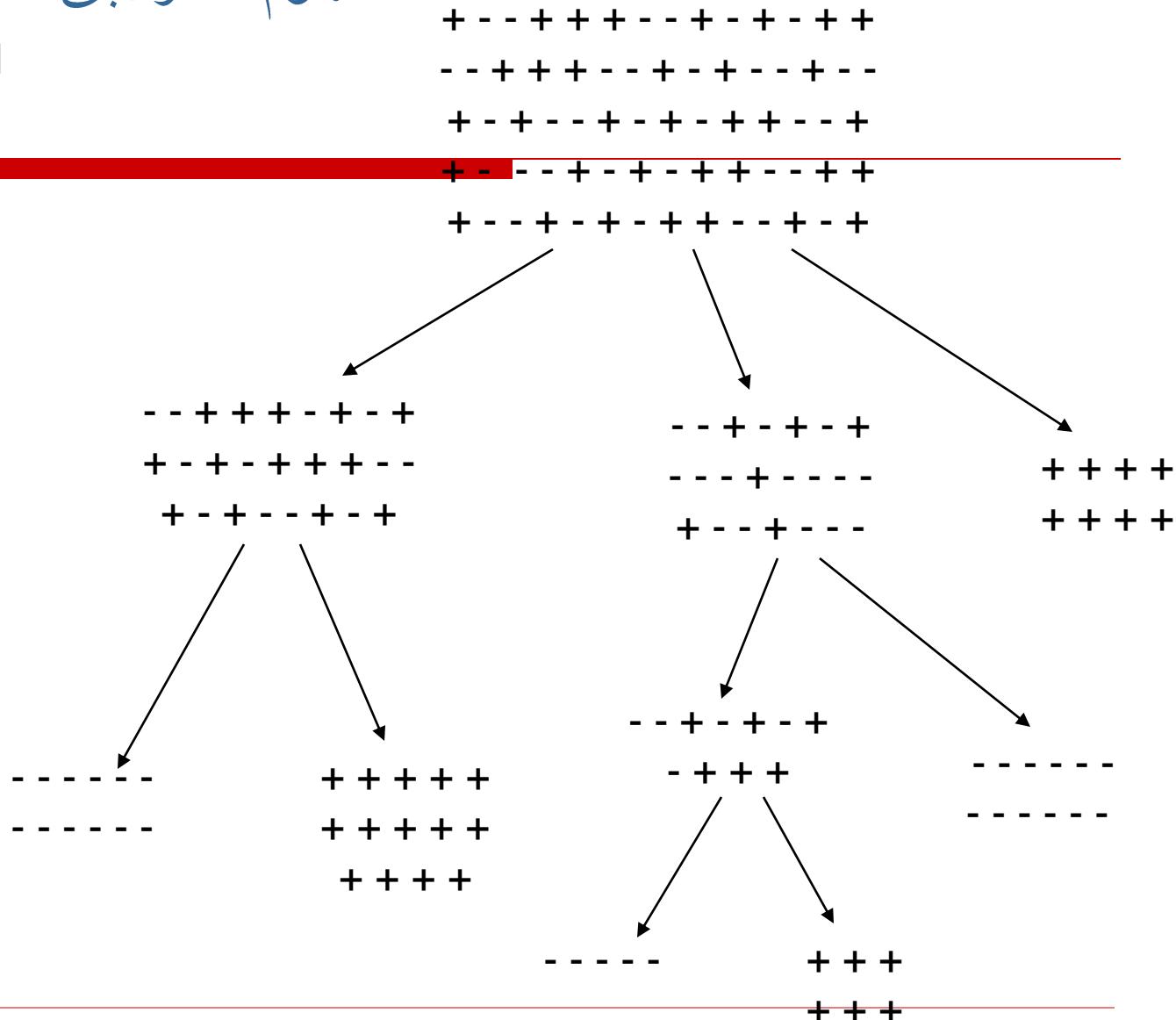
مفهوم آنتروپی

Highly Disorganized

High Entropy

Highly Organized

Low Entropy



آنتروپی

- یک تعبیر آنتروپی از تئوری اطلاعات:
 - مینیمم تعداد بیت‌هایی از اطلاعات که لازم است برای طبقه‌بندی یک عضو دلخواه از S کد شود.
 - اگر $p_+ = 1$: گیرنده می‌داند که مثال انتخاب شده + است.
 - لازم نیست پیغامی فرستاده شود: $\text{Entropy} = 0$
 - اگر $p_+ = 0.5$: ۱ بیت باید فرستاده شود تا + یا - بودن مثال مشخص شود: $\text{Entropy} = 1$
 - اگر $p_+ = 0.8$: کد کردن مجموعه‌ای از پیغام‌ها با کمتر از ۱ بیت در هر پیغام
 - تخصیص کدهای کوتاه‌تر به مثال‌های + و کدهای بلندتر به مثال‌های -

تعریف آنتروپی در حالت کلی

ویژگی هدف می‌تواند C مقدار متفاوت بگیرد: □

c-wise classification

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

$$\text{Max } (Entropy) = \log_2 c$$

آنتروپی شرطی خاص

- Specific Conditional Entropy:
- $H(Y | X=v)$ = The entropy of Y among only those records in which X has value v

$$= - \sum_{j=1}^n P(Y = j | X = v) \log_2 P(Y = j | X = v)$$

آنتروپی شرطی

- Conditional Entropy:
- $H(Y | X)$ = The average specific conditional entropy of Y
 - = Expected number of bits to transmit Y if both sides will know the value of X
 - = $\sum_j P(X = v_j)H(Y | X = v_j)$

بهره اطلاعات information gain

$IG(Y|X)$: می خواهیم Y را بفرستیم. اگر هر دو طرف X را بدانند، چه تعداد بیت بطور متوسط در ارسال صرفه جویی می شود.

واضح است که به اندازه $H(Y|X)$ از Entropy(Y) کم می شود:

$$IG(Y|X) = H(Y) - H(Y | X)$$

بهره اطلاعات information gain

مثال: 

wealth values: poor rich

gender Female 14423 1769 |  $H(\text{wealth} | \text{gender} = \text{Female}) = 0.497654$

Male 22732 9918 |  $H(\text{wealth} | \text{gender} = \text{Male}) = 0.885847$

$H(\text{wealth}) = 0.793844$ $H(\text{wealth}|\text{gender}) = 0.757154$

$IG(\text{wealth}|\text{gender}) = 0.0366896$

اندازه‌گیری مقدار قابل انتظار آنتروپی با بهره اطلاعات

□ بهره اطلاعات: معیار اندازه‌گیری مفید بودن یک ویژگی در طبقه‌بندی داده‌های آموزشی

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

$$S_v = \{s \in S | A(s) = v\}$$

□: تعداد بیتی که با داشتن مقدار ویژگی A در کد کردن مقدار هدف از یک عضو دلخواه S ذخیره می‌شود.

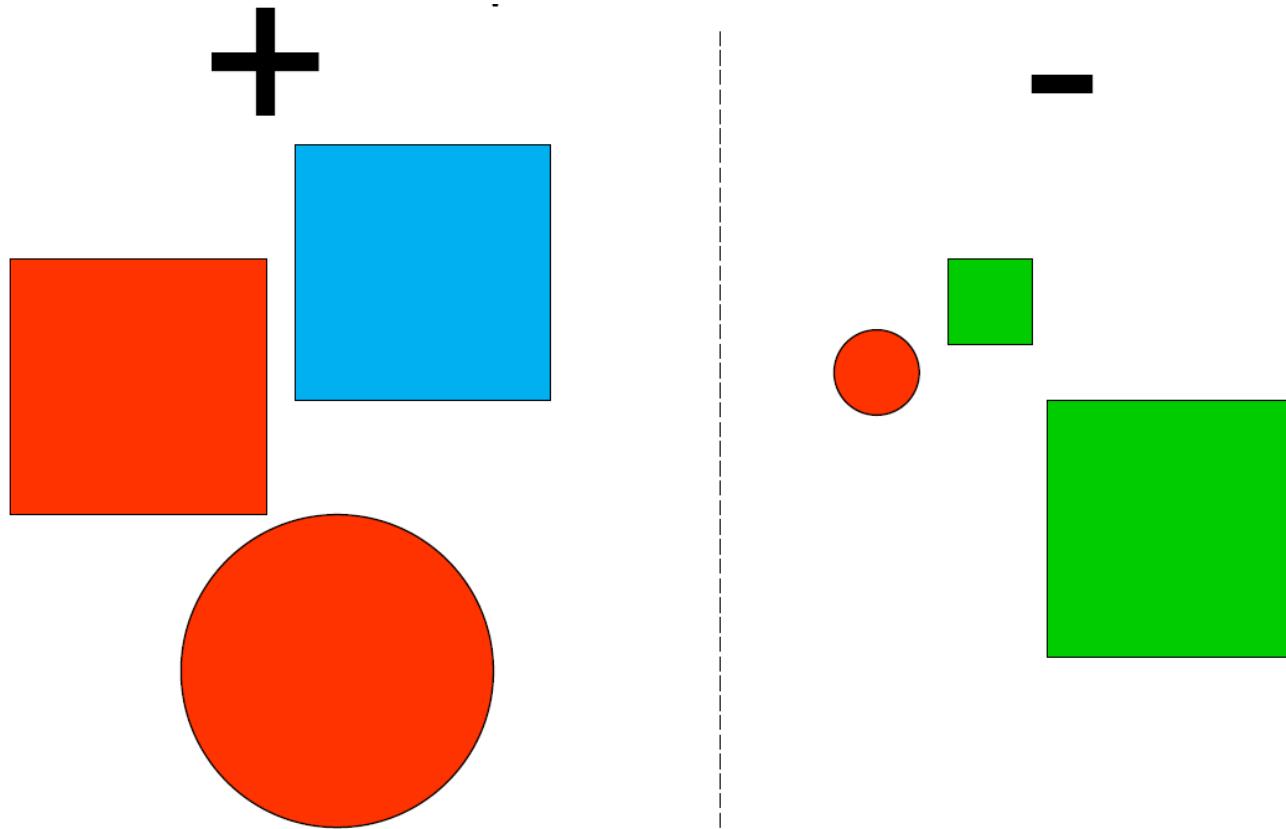
اندازه‌گیری مقدار قابل انتظار آنتروپی با بھرہ اطلاعات

- الگوریتم ID3:
- معیار اندازه‌گیری انتخاب بهترین ویژگی در هر گام در زمان
- رشد درخت: **information gain**

محاسبه بھرہ اطلاعات information gain

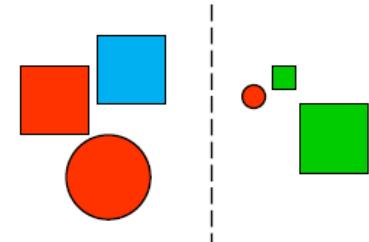
Features: color, shape, size

: مثال 



محاسبه بھرہ اطلاعات -information gain مثال

Example	Color	Shape	Size	Class
1	Red	Square	Big	+
2	Blue	Square	Big	+
3	Red	Circle	Big	+
4	Red	Circle	Small	-
5	Green	Square	Small	-
6	Green	Square	Big	-

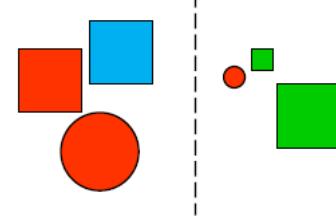


$$H(class) =$$

$$H(class | color) =$$

محاسبه بھرہ اطلاعات -information gain

Example	Color	Shape	Size	Class
1	Red	Square	Big	+
2	Blue	Square	Big	+
3	Red	Circle	Big	+
4	Red	Circle	Small	-
5	Green	Square	Small	-
6	Green	Square	Big	-



$$H(\text{class}) = H(3/6, 3/6) = 1$$

$$H(\text{class} | \text{color}) = 3/6 * H(2/3, 1/3) + 1/6 * H(1, 0) + 2/6 * H(0, 1)$$

3 out of 6
are red

2 of the
red are +

1 out of 6
is blue

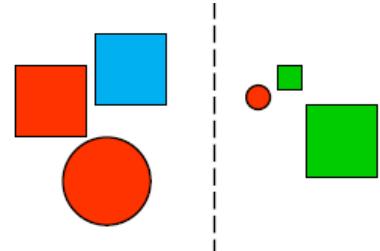
blue is +

2 out of 6
are green

green is -

محاسبه بھرہ اطلاعات -information gain

Example	Color	Shape	Size	Class
1	Red	Square	Big	+
2	Blue	Square	Big	+
3	Red	Circle	Big	+
4	Red	Circle	Small	-
5	Green	Square	Small	-
6	Green	Square	Big	-



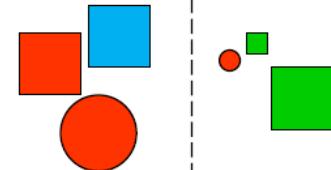
$$H(\text{class}) = H(3/6, 3/6) = 1$$

$$H(\text{class} | \text{color}) = 3/6 * H(2/3, 1/3) + 1/6 * H(1,0) + 2/6 * H(0,1)$$

$$I(\text{class}; \text{color}) = H(\text{class}) - H(\text{class} | \text{color}) = 0.54 \text{ bits}$$

محاسبه بھرہ اطلاعات -information gain

Example	Color	Shape	Size	Class
1	Red	Square	Big	+
2	Blue	Square	Big	+
3	Red	Circle	Big	+
4	Red	Circle	Small	-
5	Green	Square	Small	-
6	Green	Square	Big	-



$$H(\text{class}) = H(3/6, 3/6) = 1$$

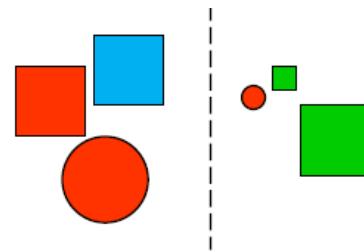
$$H(\text{class} | \text{shape}) = 4/6 * H(1/2, 1/2) + 2/6 * H(1/2, 1/2)$$

$$I(\text{class}; \text{shape}) = H(\text{class}) - H(\text{class} | \text{shape}) = 0 \text{ bits}$$

Shape tells us
nothing about
the class!

محاسبه بھرہ اطلاعات -information gain

Example	Color	Shape	Size	Class
1	Red	Square	Big	+
2	Blue	Square	Big	+
3	Red	Circle	Big	+
4	Red	Circle	Small	-
5	Green	Square	Small	-
6	Green	Square	Big	-



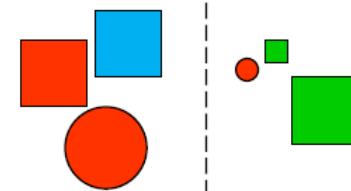
$$H(\text{class}) = H(3/6, 3/6) = 1$$

$$H(\text{class} | \text{size}) = 4/6 * H(3/4, 1/4) + 2/6 * H(0,1)$$

$$I(\text{class}; \text{size}) = H(\text{class}) - H(\text{class} | \text{size}) = 0.46 \text{ bits}$$

محاسبه بھرہ اطلاعات -information gain

Example	Color	Shape	Size	Class
1	Red	Square	Big	+
2	Blue	Square	Big	+
3	Red	Circle	Big	+
4	Red	Circle	Small	-
5	Green	Square	Small	-
6	Green	Square	Big	-



$$I(\text{class}; \text{color}) = H(\text{class}) - H(\text{class} | \text{color}) = 0.54 \text{ bits}$$

$$I(\text{class}; \text{shape}) = H(\text{class}) - H(\text{class} | \text{shape}) = 0 \text{ bits}$$

$$I(\text{class}; \text{size}) = H(\text{class}) - H(\text{class} | \text{size}) = 0.46 \text{ bits}$$

→ We select **color** as the question at root

محاسبه بھرہ اطلاعات -information gain مثال

S:[9+,5-] :PlayTennis در مثال ۱۴

Day	Outlook	Temperature	Humidity	Wind	PlayTennis?
1	Sunny	Hot	High	Light	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Light	Yes
4	Rain	Mild	High	Light	Yes
5	Rain	Cool	Normal	Light	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Light	No
9	Sunny	Cool	Normal	Light	Yes
10	Rain	Mild	Normal	Light	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Light	Yes
14	Rain	Mild	High	Strong	No

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

[Mitchell 97]

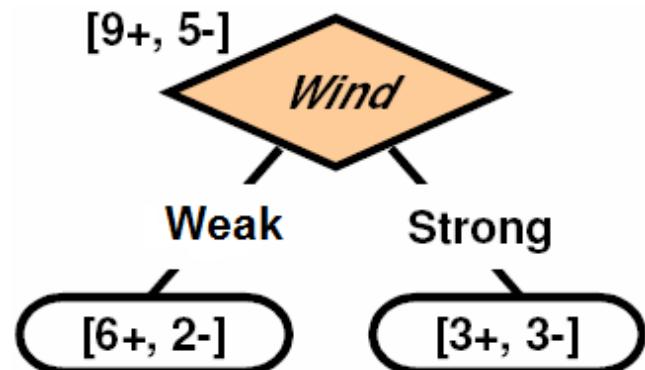
محاسبه بهره اطلاعات -information gain

$Values(Wind) = Weak, Strong$

$$S = [9+, 5-]$$

$$S_{Weak} \leftarrow [6+, 2-]$$

$$S_{Strong} \leftarrow [3+, 3-]$$



$$\begin{aligned}
 Gain(S, Wind) &= Entropy(S) - \sum_{v \in \{Weak, Strong\}} \frac{|S_v|}{|S|} Entropy(S_v) \\
 &= Entropy(S) - (8/14)Entropy(S_{Weak}) \\
 &\quad - (6/14)Entropy(S_{Strong}) \\
 &= 0.940 - (8/14)0.811 - (6/14)1.00 \\
 &= 0.048
 \end{aligned}$$

محاسبه بهره اطلاعات -information gain

$Values(Humidity) = High, Normal$

$$S = [9+, 5-]$$

$$S_{High} \leftarrow [3+, 4-]$$

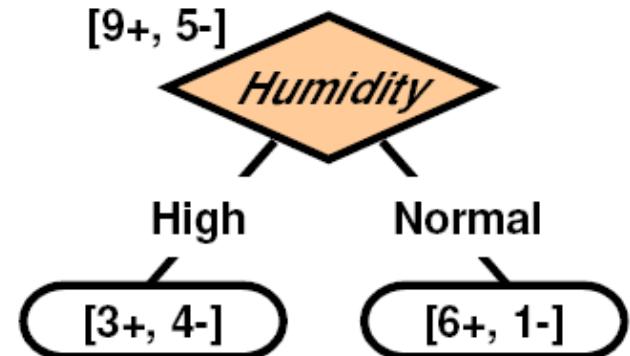
$$S_{Normal} \leftarrow [6+, 1-]$$

$$Gain(S, Humidity) = Entropy(S) - \frac{7}{14} Entropy(S_{High}) - \frac{7}{14} Entropy(S_{Normal})$$

$$Entropy(S_{High}) = -\frac{3}{7} \log_2 \frac{3}{7} - \frac{4}{7} \log_2 \frac{4}{7} = 0.985$$

$$Entropy(S_{Normal}) = -\frac{6}{7} \log_2 \frac{6}{7} - \frac{1}{7} \log_2 \frac{1}{7} = 0.592$$

$$Gain(S, Humidity) = 0.94 - \frac{7}{14} \times 0.985 - \frac{7}{14} \times 0.592 = 0.151$$



محاسبه بھرہ اطلاعات -information gain

$Values(Outlook) = Sunny, Overcast, Rain$

$$S = [9+, 5-]$$

$$S_{Sunny} \leftarrow [2+, 3-]$$

$$S_{Overcast} \leftarrow [4+, 0-]$$

$$S_{Rain} \leftarrow [3+, 2-]$$

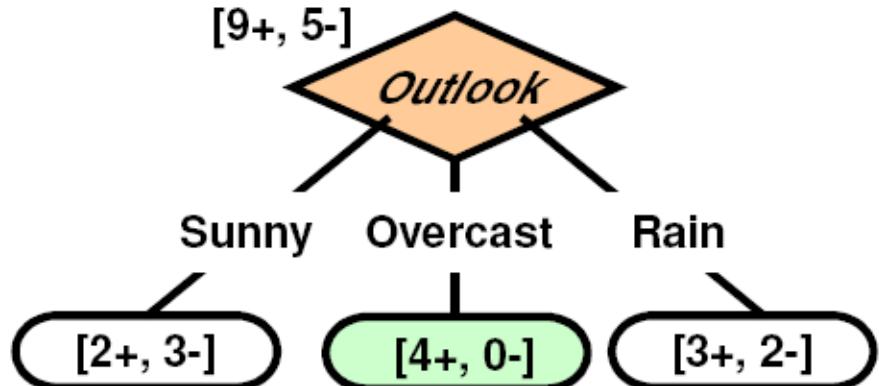
$$Gain(S, Outlook) = Entropy(S) - \frac{5}{14} Entropy(S_{Sunny}) - \frac{4}{14} Entropy(S_{Overcast}) - \frac{5}{14} Entropy(S_{Rain})$$

$$Entropy(S_{Sunny}) = -\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5} = 0.971$$

$$Entropy(S_{Overcast}) = 0$$

$$Entropy(S_{Rain}) = -\frac{3}{5} \log_2 \frac{3}{5} - \frac{2}{5} \log_2 \frac{2}{5} = 0.971$$

$$Gain(S, Outlook) = 0.94 - \frac{5}{14} \times 0.971 - \frac{4}{14} \times 0 - \frac{5}{14} \times 0.971 = 0.246$$



محاسبه بھرہ اطلاعات -information gain مثال

Values(Temperature) = Hot, Mild, Cool

$$S = [9+, 5-]$$

$$S_{Hot} \leftarrow [2+, 2-]$$

$$S_{Mild} \leftarrow [4+, 2-]$$

$$S_{Cool} \leftarrow [3+, 1-]$$

$$Gain(S, Temp) = Entropy(S) - \frac{4}{14} Entropy(S_{Hot}) - \frac{6}{14} Entropy(S_{Mild}) - \frac{4}{14} Entropy(S_{Cool})$$

$$Entropy(S_{Hot}) = 1$$

$$Entropy(S_{Mild}) = -\frac{4}{6} \log_2 \frac{4}{6} - \frac{2}{6} \log_2 \frac{2}{6} = 0.918$$

$$Entropy(S_{Cool}) = -\frac{3}{4} \log_2 \frac{3}{4} - \frac{1}{4} \log_2 \frac{1}{4} = 0.811$$

$$Gain(S, Temp) = 0.94 - \frac{4}{14} \times 1 - \frac{6}{14} \times 0.918 - \frac{4}{14} \times 0.811 = 0.029$$

محاسبه بهره اطلاعات و انتخاب گره ریشه - مثال

$$Gain(S, Outlook) = 0.246$$

$$Gain(S, Humidity) = 0.151$$

$$Gain(S, Wind) = 0.048$$

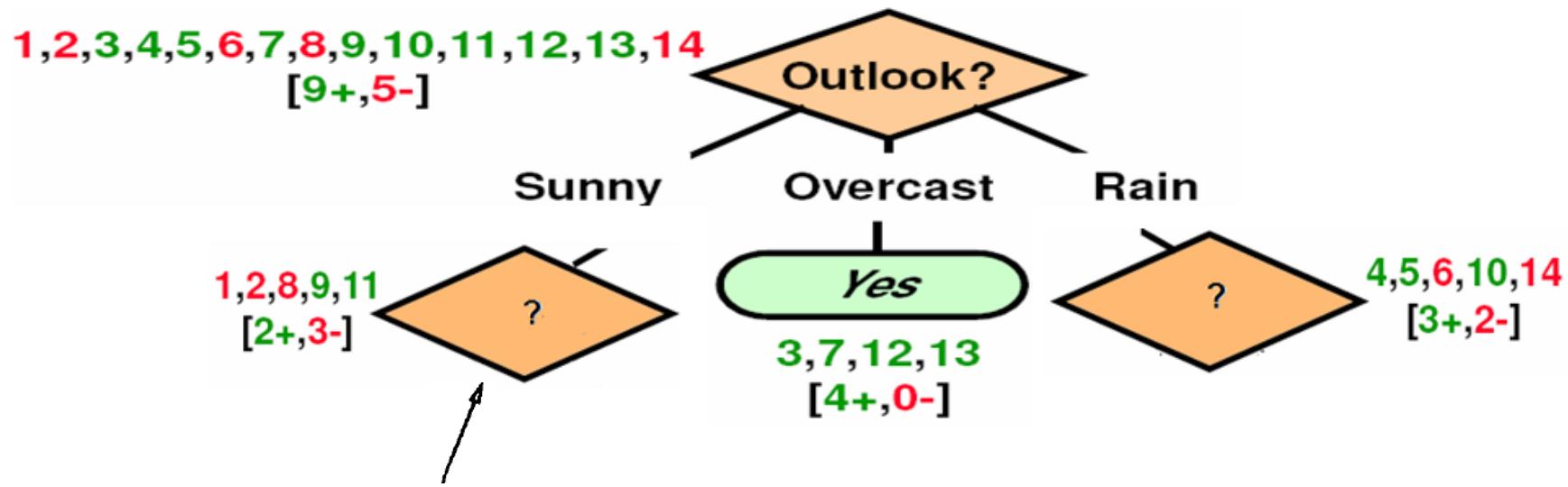
$$Gain(S, Temperature) = 0.029$$

□ بیشترین مقدار IG: مربوط به ویژگی Outlook

■ قرار گرفتن Outlook در گره ریشه

■ شاخه‌های گره ریشه: Sunny, Overcast, Rain

محاسبه بهره اطلاعات و انتخاب بهترین ویژگی – مثال



$$S_{sunny} = \{D1, D2, D8, D9, D11\}$$

$$Gain(S_{sunny}, \text{Humidity}) = .970 - (3/5) 0.0 - (2/5) 0.0 = .970$$

$$Gain(S_{sunny}, \text{Temperature}) = .970 - (2/5) 0.0 - (2/5) 1.0 - (1/5) 0.0 = .570$$

$$Gain(S_{sunny}, \text{Wind}) = .970 - (2/5) 1.0 - (3/5) .918 = .019$$

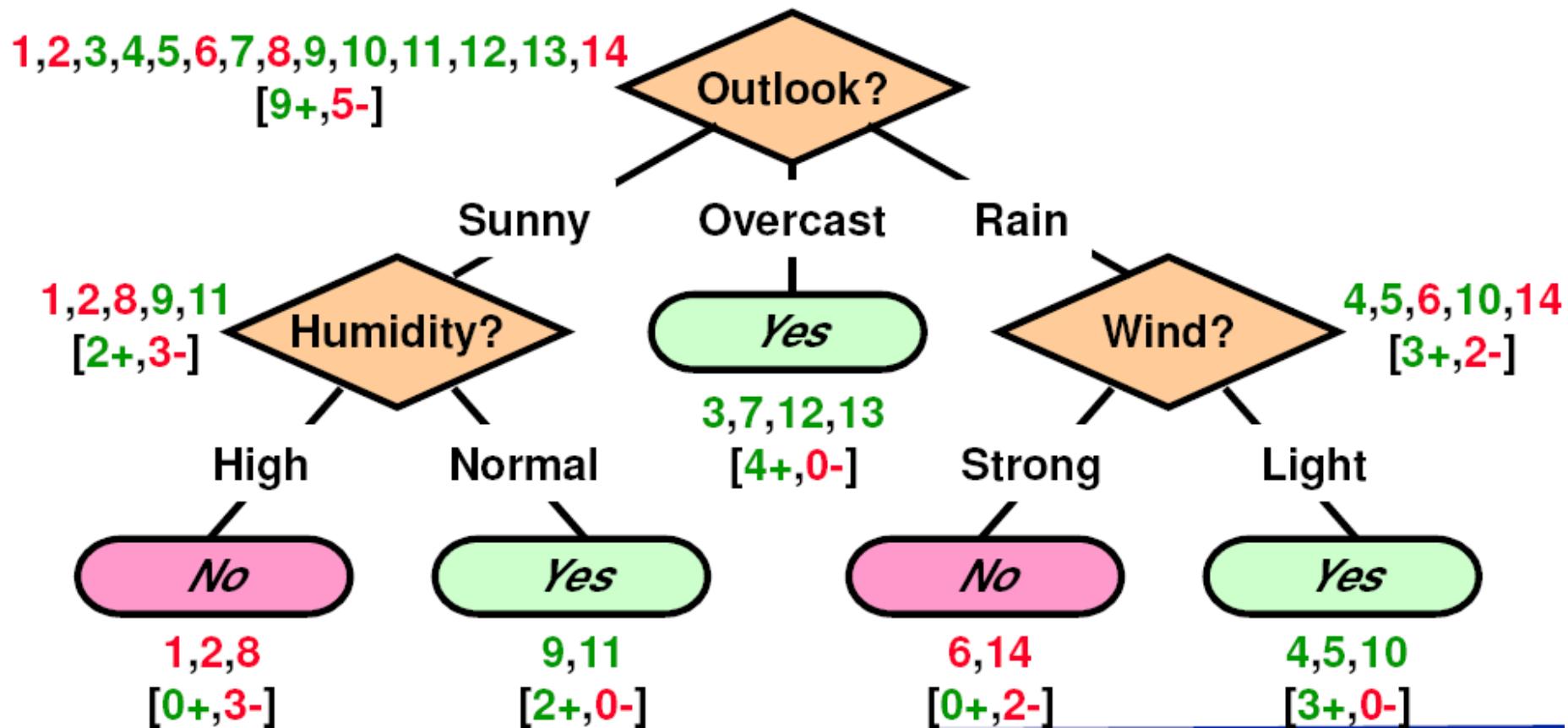
[Mitchell 97]

⁶⁰ Fig. 3.4. The partially learned decision tree resulting from the first step of ID3.

محاسبه بهره اطلاعات و انتخاب بهترین ویژگی

- ادامه پروسه انتخاب ویژگی و پارسیشن‌بندی مثال‌های آموزشی برای هر گره میانی: فقط با مثال‌های آموزشی مربوط به آن گره
 - حذف ویژگی‌هایی که در بخش‌های بالاتر درخت بکار رفته‌اند.
 - هر ویژگی حداکثر فقط ۱ بار در هر مسیری از درخت ظاهر می‌شود.
- ادامه پروسه تا برآورده شدن یکی از شرایط:
 - همه ویژگی‌ها در آن مسیر درخت بکار رفته باشند.
 - تمام مثال‌های آموزشی مربوط به این گره برگ، مقدار ویژگی هدف یکسان داشته باشند. ($\text{Entropy}=0$)

درخت تصمیم نهایی مثال PlayTennis با ID3



جستجوی فضای فرضیه در یادگیری درخت تصمیم

- ID3: بصورت جستجوی فضای فرضیه‌ها برای فرضیه‌ای که با مثال‌های آموزشی سازگار باشد. کدام فضای فرضیه‌ها؟
 - مجموعه درخت‌های تصمیم ممکن
 - انجام یک جستجوی ساده به پیچیده hill-climbing از میان این فضای فرضیه‌ها
 - شروع از یک درخت تهی و درنظر گرفتن فرضیه‌های جزئی‌تر در راستای جستجوی درخت تصمیمی که بطور صحیح داده‌های آموزشی را طبقه‌بندی می‌کند.
 - تابع ارزیابی هدایت‌گر جستجو: معیار اندازه‌گیری information gain

جستجوی فضای

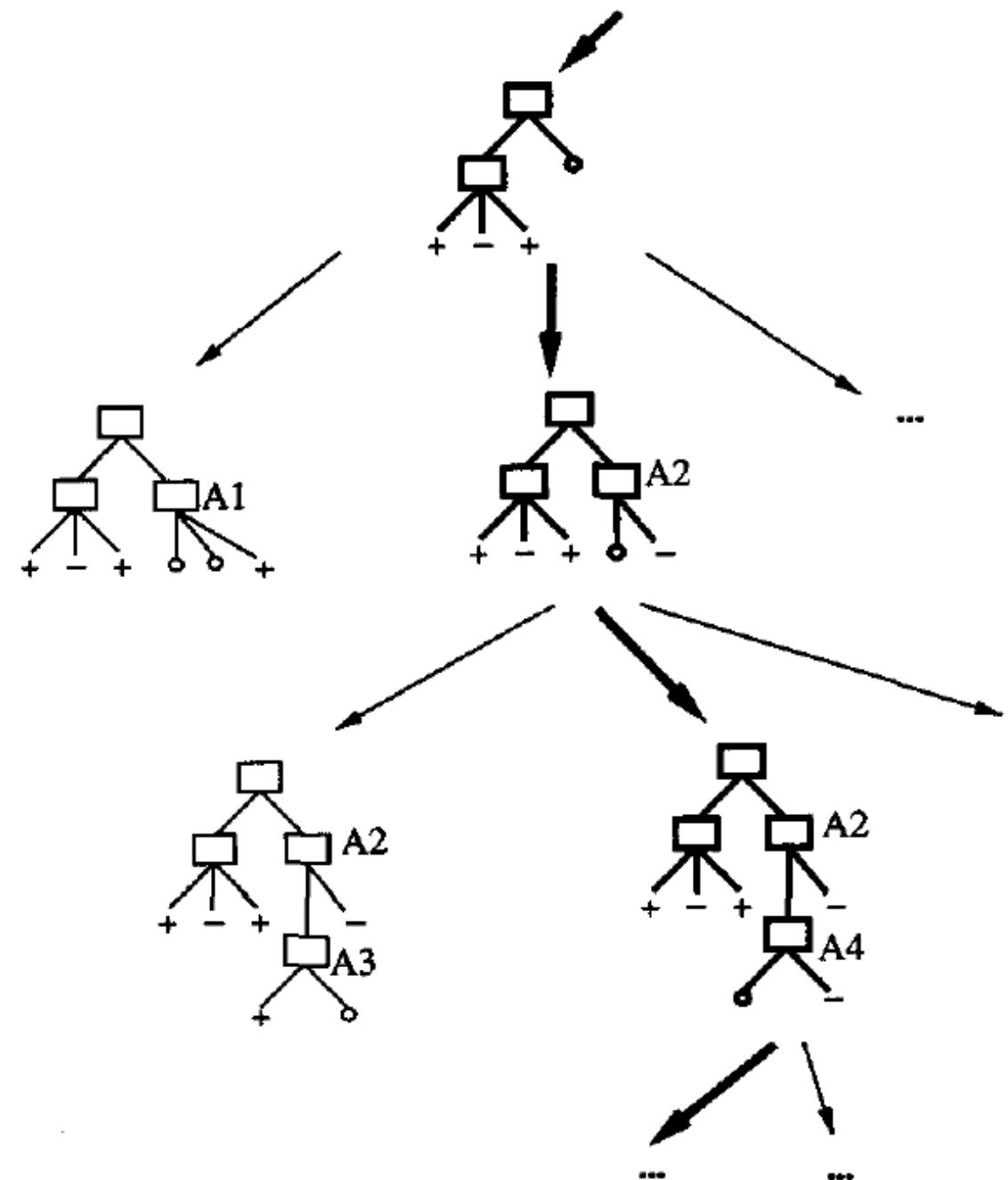
فرضیه در

یادگیری درخت

تصمیم

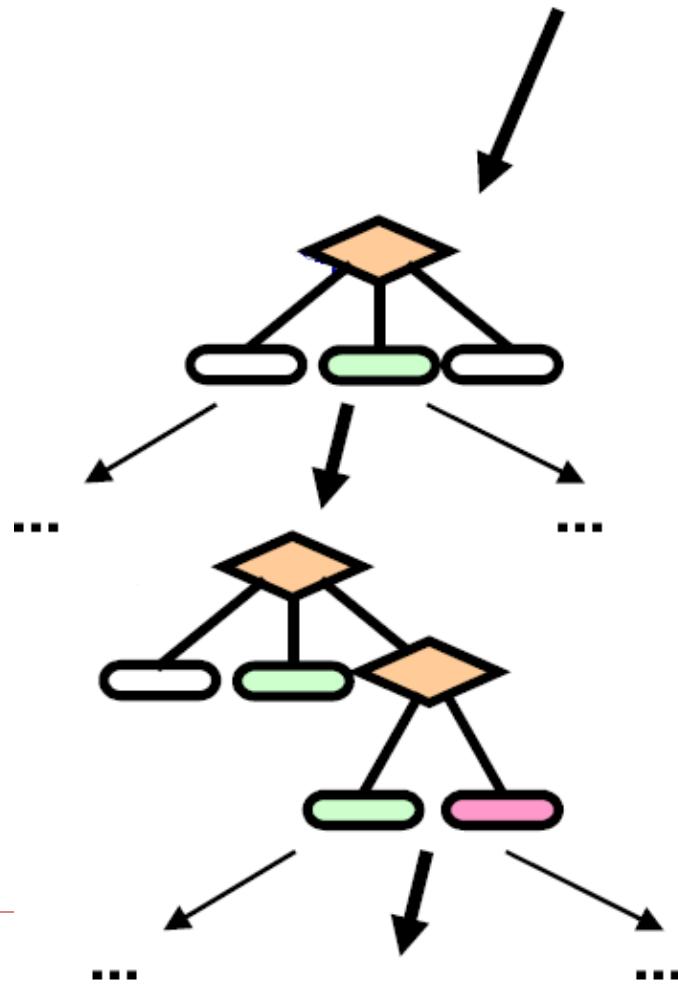
FIGURE 3.5

Hypothesis space search by ID3.
ID3 searches through the space of possible decision trees from simplest to increasingly complex, guided by the information gain heuristic.



[Mitchell 97]

جستجوی فضای فرضیه در یادگیری درخت تصمیم



جستجوی فضای فرضیه در یادگیری درخت تصمیم

□ توانایی‌ها و محدودیت‌های ID3:

- ❖ فضای فرضیه جستجوی تمام درخت‌های تصمیم ID3: یک فضای کامل از توابع با مقدار گستره و محدود مرتبط با ویژگی‌های در دسترس
- ❖ عدم وجود یکی از ریسک‌های اصلی روش‌هایی که جستجوی غیرکامل فضای فرضیه‌ها را انجام می‌دهند: عدم وجود این ریسک که فضای فرضیه‌ها ممکن است شامل تابع هدف نباشند.

جستجوی فضای فرضیه در یادگیری درخت تصمیم

- حفظ فقط یک فرضیه موجود تکی در جستجو در فضای درخت‌های تصمیم
- از دست دادن توانایی‌هایی که از نمایش همه فرضیه‌های سازگار حاصل می‌شود:
- چه تعداد درخت تصمیم جایگزین سازگار با داده‌های آموزشی وجود دارد.
- نمی‌تواند query‌های جدید که تعداد فرضیه‌های رقیب را کم می‌کند ارائه دهد.

جستجوی فضای فرضیه در یادگیری درخت تصمیم

- عدم وجود هیچ عقب‌گرد در جستجو در شکل خالص ID3
- حساس به ریسک‌های متداول جستجوی hill-climbing بدون عقب‌گرد: همگرا شدن به راه حل‌های اپتیمم محلی
- ❖ امکان اضافه کردن فرمی از عقب‌گرد در گسترش الگوریتم با post-pruning
- ❖ استفاده از تمام داده‌های آموزشی در هر گام از جستجو و تصمیم بر مبنای آمار:
- ❖ مزیت استفاده از ویژگیهای آماری همه مثال‌ها: کمتر حساس بودن جستجوی نتیجه به خطاهای موجود در تک تک مثال‌های آموزشی

بایاس استقراء در یادگیری درخت تصمیم

- سیاستی که توسط آن ID3 می‌تواند از مثال‌های آموزشی مشاهده شده به طبقه‌بندی نمونه‌های مشاهده نشده تعمیم یابد؟
- بایاس استقراء: مجموعه‌ای از فرض‌ها که به همراه داده‌های آموزشی، طبقه‌بندی نمونه‌های آینده را بصورت استقرایی توجیه می‌کند.
- توصیف بایاس استقراء ID3: توصیف پایه‌ای که توسط آن ID3 یکی از درخت‌های تصمیم را در میان بقیه انتخاب می‌کند.

بایاس استقراء در یادگیری درخت تصمیم

- بایاس استقراء تقریبی ID3: ترجیح دادن درخت‌های کوتاه‌تر به بلند‌تر
- یک تقریب نزدیکتر از بایاس استقراء در ID3: درخت‌های کوتاه‌تر به بلند‌تر ترجیح داده می‌شوند. درخت‌هایی که ویژگی‌های با بهره اطلاعاتی بیشتر را نزدیک به ریشه قرار می‌دهند، به آنها این کار را نمی‌کنند ترجیح دارند.

بایاس‌های Preference و Restriction

- وجود یک تفاوت جالب بین بایاس استقراء ID3 و الگوریتم حذف کاندید: تفاوت جستجوی فضای فرضیه ←
- :ID3 ■
- انجام یک جستجو در فضای فرضیه کامل (توانایی بیان هرتابع با مقدار گستته محدود)
- انجام یک جستجوی غیرکامل در این فضا از فرضیه‌های ساده به پیچیده تا برقراری شرط پایانی
- بایاس استقراء: نتیجه مرتب کردن فرضیه‌ها توسط استراتژی جستجوی آن. و گرنه فضای فرضیه آن هیچ بایاس اضافی ندارد.

بایاس‌های Preference و Restriction

- الگوریتم حذف کاندید VS:
- انجام یک جستجو در فضای فرضیه ناکامل (فقط زیرمجموعه‌ای از مفاهیم قابل آموزش را بیان می‌کند.)
- جستجوی کامل این فضا و پیدا کردن هر فرضیه سازگار با داده‌های آموزشی
- بایاس استقراء: نتیجه توان بیان نمایش فرضیه‌ها توسط آن. و گرنه استراتژی جستجوی آن هیچ بایاس اضافی ندارد.

بایاس‌های Preference و Restriction

- بایاس استقراء ID3: حاصل از استراتژی جستجوی آن یک ترجیح برای فرضیه‌های خاص نسبت به دیگران (مثلاً برای a preference bias (search bias) : فرضیه‌های کوتاه‌تر):
- بایاس استقراء الگوریتم حذف کاندید: حاصل از تعریف فضای جستجوی آن.
- یک محدودیت روی مجموعه فرضیه‌های درنظر گرفته شده: a restriction bias (language bias)

بایاس‌های Preference و Restriction

- کدام نوع بایاس ترجیح دارد؟
- معمولاً بایاس **preference** مطلوب‌تر است. زیرا اجازه می‌دهد که یادگیر در یک فضای فرضیه کامل که با اطمینان تابع هدف مجهول را شامل می‌شود، کار کند.
- بعضی سیستم‌های یادگیری هر دو بایاس را ترکیب می‌کنند.
- مثال: برنامه فصل ۱ برای یادگیری یک تابع ارزیابی عددی برای بازی **restriction** استفاده از تابع خطی برای نمایش تابع ارزیابی: بایاس **preference**: بایاس LMS
- انتخاب روش تنظیم پارامترها با

نکاتی در یادگیری درخت تصمیم

- چقدر در رشد درخت تصمیم می‌توانیم جلو برویم؟
- کار کردن با ویژگیهای با مقدار پیوسته
- انتخاب یک معیار اندازه‌گیری مناسب برای انتخاب ویژگی
- کار کردن با داده‌های آموزشی با مقدار ویژگی گم شده
- کار کردن با ویژگیهای با هزینه‌های (cost) متفاوت

۱- پرهیز از overfit به داده‌ها

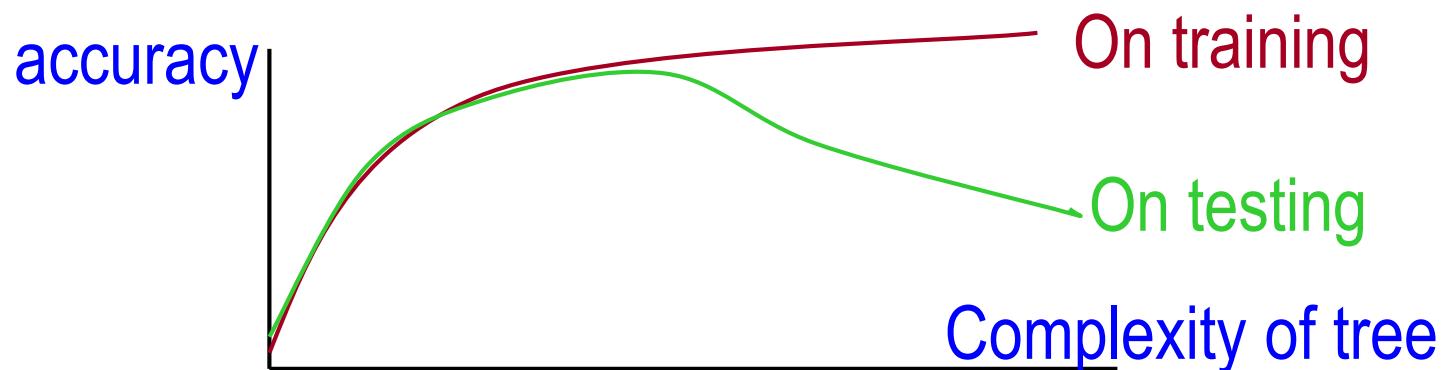
- الگوریتم ID3: رشد هر شاخه درخت تا مثال‌های آموزشی را بطور کامل طبقه‌بندی کند.
- در حالات زیر این الگوریتم می‌تواند درخت‌هایی تولید کند که به مثال‌های آموزشی overfit شوند:
 - داده‌ها نویز داشته باشند.
 - تعداد مثال‌های آموزشی برای تولید یک نمونه تابع هدف صحیح خیلی کم باشد.
- مثال: اگر فقط دو بار پرتاپ سکه داشته باشیم، و هر ۲ بار شیر آمده باشد چه نتیجه‌ای در مورد این آزمایش می‌توان گرفت؟

۱- پرهیز از overfit به داده‌ها

- تعریف: یک فضای فرضیه H داریم. یک فرضیه $h \in H$ به داده‌های آموزشی overfit می‌شود، اگر فرضیه جایگزین دیگری به نام $h' \in H$ وجود داشته باشد که:
- h روی مثال‌های آموزشی خطای کمتری از h' دارد.
- ولی h' خطای کوچکتری از h روی کل توزیع نمونه‌ها (شامل داده‌های خارج از مجموعه آموزشی) دارد.
- h' کمتر از h با داده‌های آموزشی سازگار است.

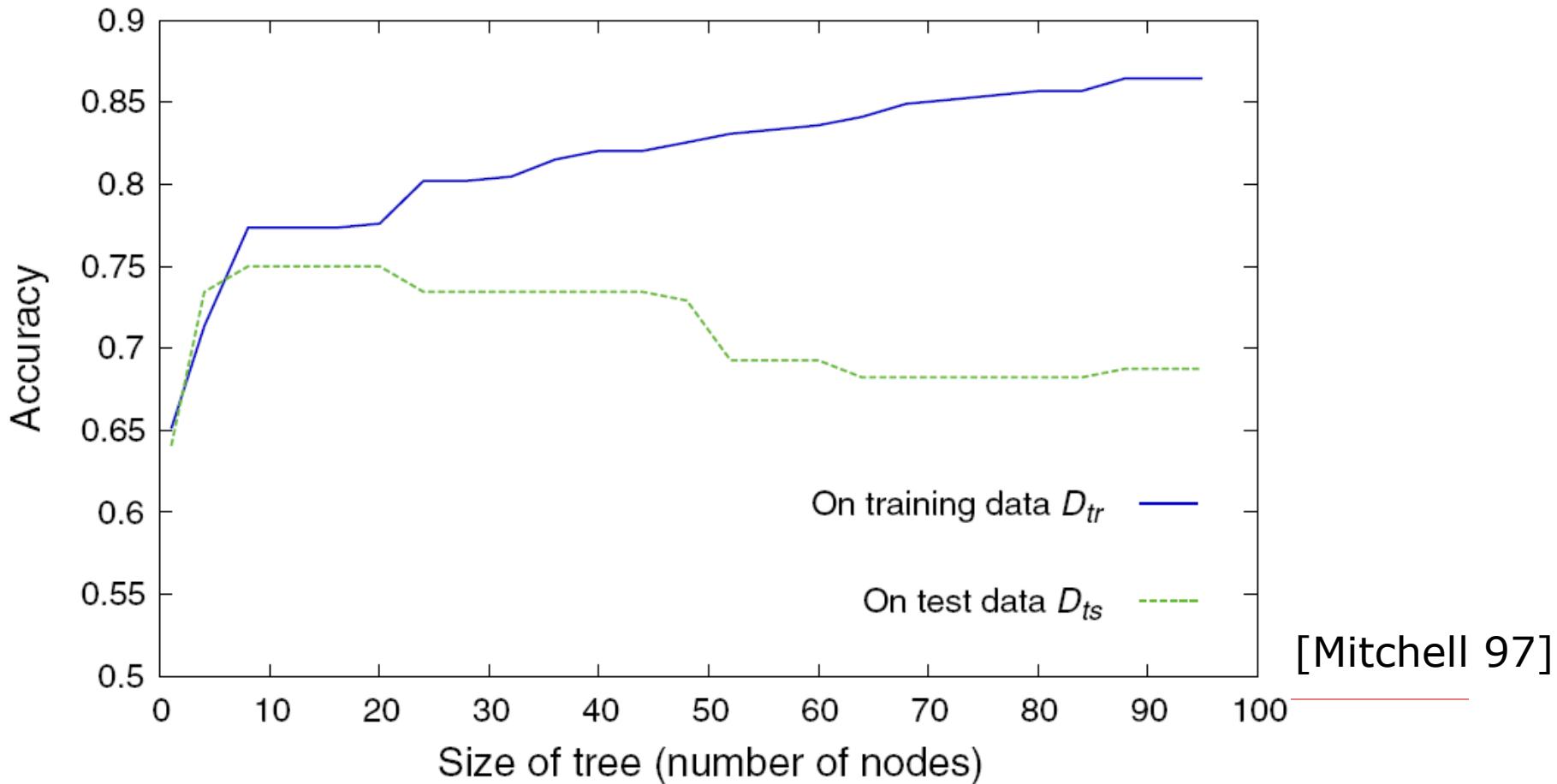
۱- پرهیز از overfit به داده‌ها

□ چگونگی افزایش صحت تصمیم با افزایش تعداد گره‌ها (پیچیده‌تر شدن درخت) در دادگان آموزشی و افزایش و سپس کاهش آن در دادگان تست:



۱- پرهیز از overfit به داده‌ها

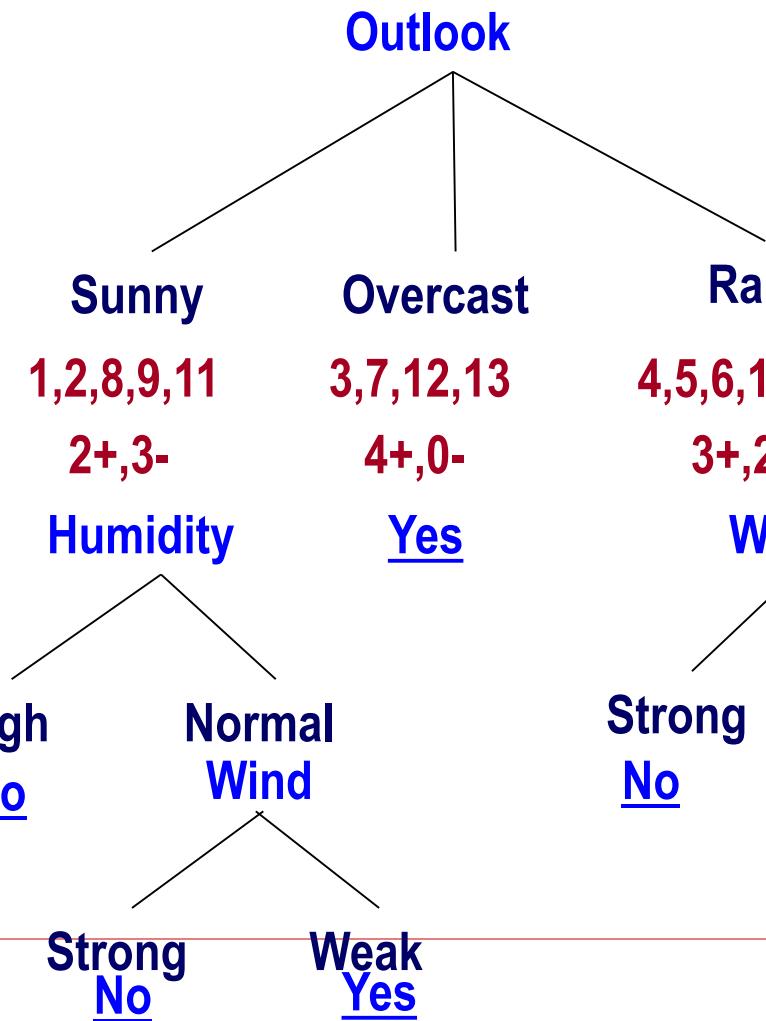
مثال: یادگیری اینکه کدام بیماران فرمی از دیابت دارند. □



۱- پرهیز از overfit به داده‌ها

- چرا یک درخت h می‌تواند بهتر از ' h' به مثال‌های آموزشی fit شود، ولی روی مثال‌های دیگر بدتر عمل کند؟
- یک دلیل: زمانی که مثال‌های آموزشی خطاهای تصادفی یا نویز داشته باشند.
- مثال: اثر افزایش مثال آموزشی + زیر که به اشتباه منفی برچسب خورده به ۱۴ مثال :PlayTennis
- <Outlook=Sunny,Temp=Hot, Humidity=Normal, Wind=Strong,
PlayTennis=NO>
- پیچیده‌تر شدن درخت overfit \leftarrow ID3

۱- پرهیز از overfit به داده‌ها



□ مثال: درخت جدید شده به مثال‌های آموزشی با افزودن داده نویزی در مثال overfit PlayTennis

۱- پرهیز از overfit به داده‌ها

- رویکردهای متعدد برای جلوگیری از overfitting در یادگیری درخت تصمیم، به ۲ گروه تقسیم می‌شوند:
 - توقف رشد درخت قبل از اینکه درخت به نقطه‌ای برسد که بطور کامل مثال‌های آموزشی را طبقه‌بندی کند.
 - اجازه دادن اینکه درخت به داده‌ها overfit شود، و بعد هرس کردن (post-prune) آن.
- موفق‌تر بودن رویکرد دوم در عمل:
 - چون در رویکرد اول، تخمین اینکه چه زمانی رشد درخت را متوقف کنیم، سخت است.

۱- پرهیز از overfit به داده‌ها

□ معیارهای تعیین سایز صحیح درخت:

- ۱- استفاده از یک مجموعه متفاوت از مثال‌های آموزشی و انجام ارزیابی برای post-pruning روی آن
- ۲- استفاده از کل داده‌های در دسترس برای آموزش، و انجام یک تست آماری برای تخمین اینکه توسعه یک گره خاص به بهبودی به خارج از مثال‌های آموزشی می‌رسد یا هرس آن.
- ۳- استفاده از یک معیار اندازه‌گیری صریح پیچیدگی در encode کردن مثال‌های آموزشی و درخت تصمیم، و توقف رشد درخت در زمان مینیمم شدن این سایز:

بر مبنای Minimum Description Length Principle

۱- پرهیز از overfit به داده‌ها

- روش اول: متداول ترین رویکرد
- training and validation approach
- تقسیم داده در دسترس به ۲ مجموعه:

برای یادگیری فرضیه

□ training set:

برای ارزیابی صحت فرضیه بالا روی داده‌های بعدی و ارزیابی تاثیر هرس کردن این فرضیه

مجموعه validation باید خودش به اندازه کافی بزرگ باشد.

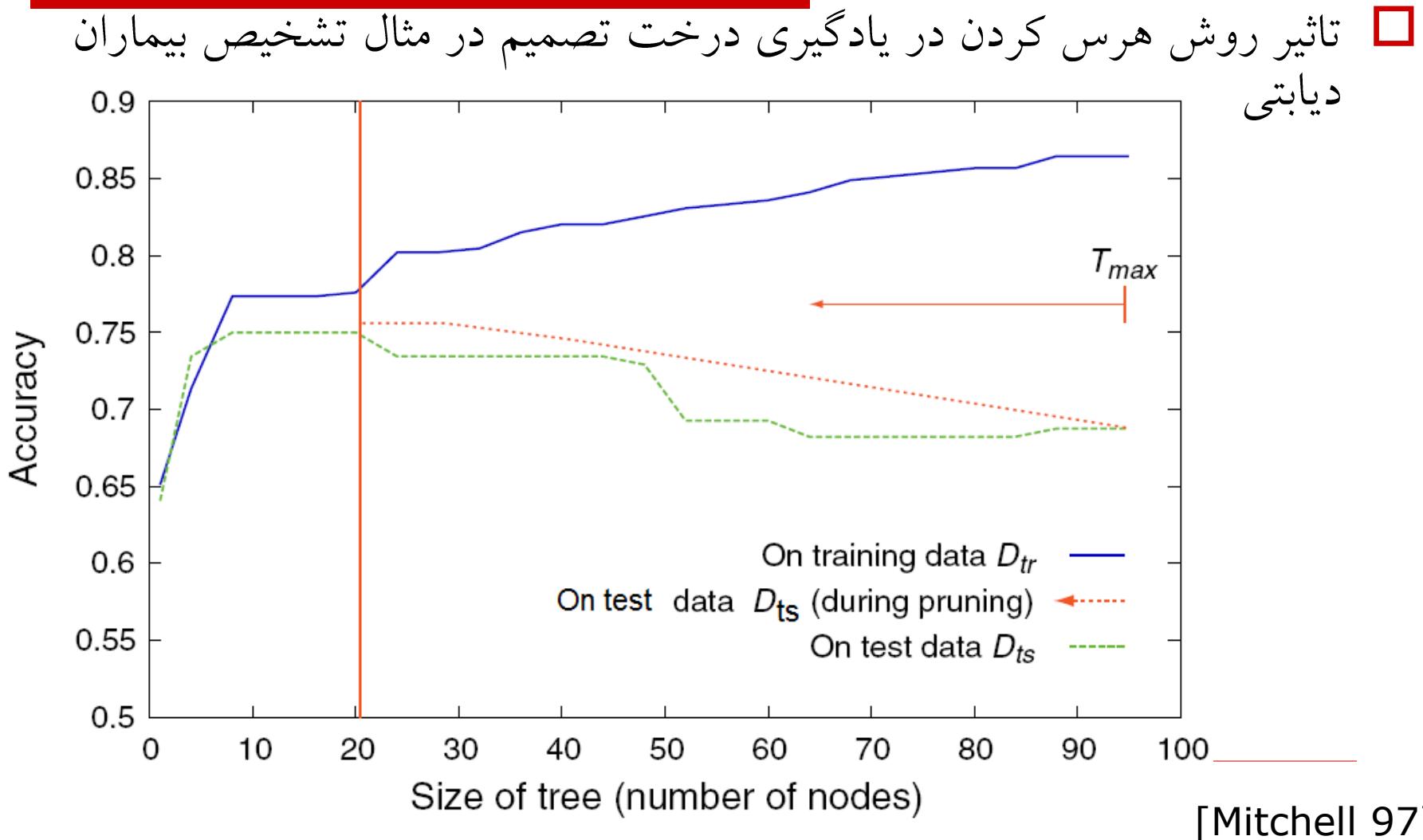
Reduced error pruning – ۱-۱

- یک رویکرد استفاده از مجموعه validation برای جلوگیری از overfitting
- در نظر گرفتن هر یک از گره‌های درخت بعنوان کاندیدای هرس کردن
- هرس کردن یک گره تصمیم:
 - حذف زیردرخت منشعب شده در آن گره، و تبدیل آن گره به گره برگ
 - تخصیص متداول‌ترین طبقه‌بندی مثال‌های آموزشی وابسته به آن گره
 - حذف گره‌ها فقط زمانی که درخت هرس شده نتیجه روی مجموعه validation بدتر از اولی عمل نکند.

Reduced error pruning - ۱-۱

- انجام هرس کردن گره‌ها بصورت تکراری:
 - ادامه دادن هرس کردن گره‌ها تا زمانی که هرس کردن بیشتر خطرناک شود (صحت درخت را روی مجموعه validation کاهش دهد).
 - مثال: همان مثال قبلی تشخیص بیماران دیابتی و بهبود صحت تصمیم روی مجموعه تست با هرس کردن
 - تقسیم داده‌ها به ۳ زیرمجموعه train, validation, test
 - شروع هرس کردن: سایز درخت ماکزیمم است، و صحت روی مجموعه تست مینیمم!

Reduced error pruning – ۱-۱



Reduced error pruning – ۱-۱

- استفاده از یک مجموعه جدا برای هدایت هرس کردن:
 - ✓ رویکرد مفید اگر داده‌های زیاد در دسترس باشد.
 - بزرگترین عیب: وقتی داده‌ها محدود است.
- استفاده از cross-validation و متوسط‌گیری نتایج
- روش rule post-pruning

Rule post-pruning - ۲-۱

- پیدا کردن درخت تصمیم از مجموعه آموزشی: رشد درخت تا زمانی که با داده‌های آموزشی تا حد ممکن سازگار باشد و با اجازه اتفاق افتادن overfitting
- تبدیل درخت یاد گرفته شده به مجموعه قوانین معادل: تولید یک قانون برای هر مسیر از گره ریشه تا گره برگ
- هرس کردن هر قانون با حذف هر پیش‌شرطی که باعث می‌شود صحت تخمینی آن بهتر شود.
- مرتب کردن قوانین هرس شده مطابق با صحت تخمینی، و استفاده از آنها در طبقه‌بندی

Rule post-pruning - ۲-۱

:PlayTennis

IF (*Outlook* = *Sunny*) \wedge (*Humidity* = *High*) THEN *PlayTennis* = *No*

- ▶ هرس کردن هر قانون با حذف هر پیششرط آن، طوری که
- ▶ حذف آن پیششرط، صحت تخمینی را بدتر نکند.
- ▶ اگر میزان صحت تخمینی کم شود، هیچ هرس کردنی انجام نمی‌شود.
- ▶ یک روش تخمین صحت: استفاده از یک مجموعه validation

۲- کار با ویژگیهای با مقدار پیوسته

□ تعریف دینامیکی ویژگیهای با مقدار گسته جدید: مقدار ویژگی پیوسته را به فواصلی از مجموعه گسته پارتیشن‌بندی می‌کنند:

□ برای یک ویژگی A با مقدار پیوسته:
■ ایجاد یک ویژگی بولی جدید A_c :

if $A < c$ True
otherwise False

■ بهترین مقدار آستانه c را چگونه انتخاب کنیم؟

۲- کار با ویژگیهای با مقدار پیوسته

انتخاب آستانه C : آنکه بیشترین information gain را تولید کند:

- مرتب کردن مثال‌ها بر طبق مقدار پیوسته A
- شناسایی مثال‌های مجاور که در طبقه‌بندی هدف فرق می‌کنند.
- تولید یک مجموعه از آستانه‌های کاندید بصورت متوسط مقادیر مربوط به A در مثال‌های مجاور
- ارزیابی آستانه‌های کاندید با محاسبه IG مرتبط با هر کدام

۲- کار با ویژگیهای با مقدار پیوسته

مثال: اضافه کردن ویژگی پیوسته دما در PlayTennis

<i>Temperature:</i>	40	48	60	72	80	90	
<i>PlayTennis:</i>	No	No	Yes	Yes	Yes	No	
$(48 + 60)/2$				$(80 + 90)/2$			

آستانه‌های کاندید: ۵۴ و ۸۵

آستانه انتخابی در نهایت: ۵۴

$$\left\{ \begin{array}{ll} \text{Temp} > 54 & \text{High} \\ \text{Temp} \leq 54 & \text{Low} \end{array} \right.$$

۳- معیارهای اندازه‌گیری دیگر برای انتخاب ویژگیها

- معیار IG : وجود یک بایاس طبیعی برای ترجیح ویژگیهایی که مقادیر زیادی دارند به آنها که مقادیر کمتری دارند.
- مثال: افزودن ویژگی “Date” که مقادیر ممکن زیادی دارد به PlayTennis
- بالاترین بهره اطلاعات را خواهد داشت و در ریشه قرار می‌گیرد، و بطور کامل داده‌های آموزشی را طبقه‌بندی می‌کند.
- اما درخت تصمیم روی مثال‌های دیگر ضعیف عمل خواهد کرد.
- مشکل این ویژگی: مقادیر ممکن زیاد و امکان جدا کردن داده‌های آموزشی به زیرمجموعه‌های خیلی کوچک — IG بالا
- اما پیشگویی کننده ضعیف برای تابع هدف روی نمونه‌های دیده نشده

۳- معیارهای اندازه‌گیری دیگر برای انتخاب ویژگیها

- راه حل: انتخاب ویژگیهای تصمیم براساس یک معیار اندازه‌گیری دیگر غیر از IG
- یک معیار موفق:
- ویژگیهایی مثل "Date" را با **SplitInformation** جرمیمه می‌کند:
 - به اینکه یک ویژگی چقدر وسیع و یکنواخت داده‌ها را تقسیم می‌کند، حساس است.

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

- در واقع آنتروپی S با توجه به مقادیر ویژگی A

۳- معیارهای اندازه‌گیری دیگر برای انتخاب ویژگیها

:SplitInformation براسas معيار Gain قبلی و GainRatio تعریف □

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

مثال: یک مجموعه از n مثال که کاملاً با ویژگی A جدا می‌شوند (مثلاً SplitInformation= $\log_2 n$ تاریخ) □

اما با یک ویژگی بولی B که همان n مثال را دقیقاً به نصف تقسیم می‌کند: □

اگر IG در A و B یکسان: B مقدار GainRatio بیشتر □

۳- معیارهای اندازه‌گیری دیگر برای انتخاب ویژگیها

- یک مساله عملی در استفاده از GainRatio به جای :Gain
- امکان صفر یا خیلی کوچک شدن مخرج، اگر $|S_i| = |S|$
- GainRatio تعریف نشده یا خیلی بزرگ برای ویژگیهایی که تقریباً
برای همه اعضاء S مقدار یکسان دارند.
- راه حل و جلوگیری از اینکه ویژگیها فقط براساس آن انتخاب شوند:
- محاسبه Gain هر ویژگی
- اگر Gain بالای یک مقدار متوسط بود: اعمال GainRatio

۴- کار با داده‌های آموزشی با مقادیر ویژگی گم شده

□ تخمین مقدار ویژگی گم شده (missing attribute value) براساس مثال‌های دیگر که این ویژگی مقدار معلوم دارد.

□ برخورد با ویژگی گم شده A در مثال آموزشی $\langle x, c(x) \rangle$ در انتخاب ویژگیها برای گره‌های درخت:

- ۱- تخصیص متداول‌ترین مقدار در مثال‌های آموزشی در گره n
- ۲- تخصیص متداول‌ترین مقدار در مثال‌های آموزشی در گره n که همان طبقه‌بندی $c(x)$ را دارند.

■ تخصیص یک احتمال به هر یک از مقادیر ممکن A به جای متداول‌ترین مقدار

۴- کار با داده‌های آموزشی با مقادیر ویژگی گم شده

Day	Outlook	Temperature	Humidity	Wind	PlayTennis	مثال:
1	Sunny	Hot	High	Weak	No	<input type="checkbox"/>
2	Sunny	Hot	High	Strong	No	<input type="checkbox"/>
8	Sunny	Mild	???	Weak	No	<input type="checkbox"/>
9	Sunny	Cool	Normal	Weak	Yes	<input type="checkbox"/>
11	Sunny	Mild	Normal	Strong	Yes	<input type="checkbox"/>

- a) the most common Humidity at Sunny
- b) as (a) but with PlayTennis = No

An example of a data set with missing attribute values

Case	Attributes			Decision
	Temperature	Headache	Nausea	
1	high	?	no	yes
2	very_high	yes	yes	yes
3	?	no	no	no
4	high	yes	yes	yes
5	high	?	yes	no
6	normal	yes	no	no
7	normal	no	yes	no
8	?	yes	?	yes

Table 1.3. Data set with missing attribute values replaced by the most common values

Case	Attributes			Decision
	Temperature	Headache	Nausea	
1	high	yes	no	yes
2	very_high	yes	yes	yes
3	high	no	no	no
4	high	yes	yes	yes
5	high	yes	yes	no
6	normal	yes	no	no
7	normal	no	yes	no
8	high	yes	yes	yes

An example of a data set with missing attribute values

داده‌های

آموزشی با

مقادیر ویژگی

گم شده - مثال

Case	Attributes			Decision
	Temperature	Headache	Nausea	
1	high	?	no	yes
2	very_high	yes	yes	yes
3	?	no	no	no
4	high	yes	yes	yes
5	high	?	yes	no
6	normal	yes	no	no
7	normal	no	yes	no
8	?	yes	?	yes

Table 1.4. Data set with missing attribute values replaced by the most common value of the attribute restricted to a concept

Case	Attributes			Decision
	Temperature	Headache	Nausea	
1	high	yes	no	yes
2	very_high	yes	yes	yes
3	normal	no	no	no
4	high	yes	yes	yes
5	high	no	yes	no
6	normal	yes	no	no
7	normal	no	yes	no
8	high	yes	yes	yes

[Grzymala
-Buss]

Table 1.7. An example of a data set with a numerical attribute

داده‌های

آموزشی با

مقادیر ویژگی

گم شده - مثال

Case	Attributes			Decision
	Temperature	Headache	Nausea	
1	100.2	?	no	yes
2	102.6	yes	yes	yes
3	?	no	no	no
4	99.6	yes	yes	yes
5	99.8	?	yes	no
6	96.4	yes	no	no
7	96.6	no	yes	no
8	?	yes	?	yes

Table 1.8. Data set in which missing attribute values are replaced by the attribute mean and the most common value

Case	Attributes			Decision
	Temperature	Headache	Nausea	
1	100.2	yes	no	yes
2	102.6	yes	yes	yes
3	99.2	no	no	no
4	99.6	yes	yes	yes
5	99.8	yes	yes	no
6	96.4	yes	no	no
7	96.6	no	yes	no
8	99.2	yes	yes	yes

[Grzymala
-Buss]

Table 1.7. An example of a data set with a numerical attribute

داده‌های

آموزشی با

مقادیر ویژگی

گم شده - مثال

Case	Attributes			Decision
	Temperature	Headache	Nausea	
1	100.2	?	no	yes
2	102.6	yes	yes	yes
3	?	no	no	no
4	99.6	yes	yes	yes
5	99.8	?	yes	no
6	96.4	yes	no	no
7	96.6	no	yes	no
8	?	yes	?	yes

Table 1.9. Data set in which missing attribute values are replaced by the attribute mean and the most common value, both restricted to the concept

Case	Attributes			Decision
	Temperature	Headache	Nausea	
1	100.2	yes	no	yes
2	102.6	yes	yes	yes
3	97.6	no	no	no
4	99.6	yes	yes	yes
5	99.8	no	yes	no
6	96.4	yes	no	no
7	96.6	no	yes	no
8	100.8	yes	yes	yes

۴- کار با داده‌های آموزشی با مقادیر ویژگی گم شده- مثال

Attribute1	Attribute2	Attribute3	Class
A	70	True	CLASS1
A	90	True	CLASS2
A	85	False	CLASS2
A	95	False	CLASS2
A	70	False	CLASS1
?	90	True	CLASS1
B	78	False	CLASS1
B	65	True	CLASS1
B	75	False	CLASS1
C	80	True	CLASS2
C	70	True	CLASS2
C	80	False	CLASS1
C	80	False	CLASS1
C	96	False	CLASS1

۴- کار با داده‌های آموزشی با مقادیر ویژگی گم شده- مثال

$$\text{Info}(T) = -\frac{8}{13}\log_2\left(\frac{8}{13}\right) - \frac{5}{13}\log_2\left(\frac{5}{13}\right) = \mathbf{0.961 \text{ bits}}$$

$$\begin{aligned}\text{Info}_{x_1}(T) &= \frac{5}{13}(-2/5\log_2(2/5) - 3/5\log_2(3/5)) \\&\quad + \frac{3}{13}(-3/3\log_2(3/3) - 0/3\log_2(0/3)) \\&\quad + \frac{5}{13}(-3/5\log_2(3/5) - 2/5\log_2(2/5)) \\&= \mathbf{0.747 \text{ bits}}\end{aligned}$$

$$\text{Gain}(x_1) = \frac{13}{14} (0.961 - 0.747) = \mathbf{0.199 \text{ bits}}$$

۵- کار با ویژگیهای با هزینه‌های متفاوت

□ مثال: طبقه‌بندی بیماری‌ها براساس ویژگیهایی مثل دما، نتایج بیوپسی، آزمایش خون،... ■ هزینه این ویژگیها متفاوت است.

□ استفاده از ویژگیهای با هزینه بالا فقط وقتی لازم است.

$$\frac{Gain(S, A)}{Cost(A)} \quad \frac{Gain^2(S, A)}{Cost(A)} \quad \frac{2^{Gain(S, A)} - 1}{(Cost(A) + 1)^w}$$

خلاصه

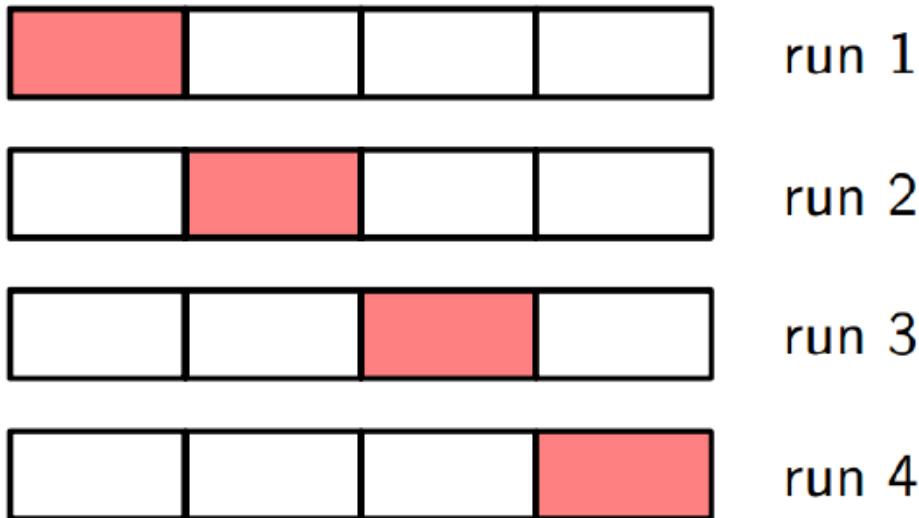
- یادگیری درخت تصمیم یک روش عملی برای یادگیری مفهوم و باقی توابع با مقدار گسته
- ID3: انجام یک جستجو در فضای فرضیه کامل (عدم وجود مشکل روشهای محدود کننده فضای فرضیه‌ها)
- با این استقراء در ID3: ترجیح در انتخاب درخت‌های کوچکتر
- مساله مهم: overfitting به داده‌های آموزشی
- روشهای post-pruning برای جلوگیری
- گسترش‌های الگوریتم ID3 برای ایجاد امکان کار با ویژگیهای با مقدار پیوسته، مقادیر گم شده،...

نکاتی در مورد انتخاب مجموعه validation برای جلوگیری از overfitting

- جدا کردن مجموعه‌ای از دادگان برای validation
- مشکل: اگر دادگان کوچک باشد: کمبود داده
- ✓ راه حل: استفاده از cross-validation
 - تقسیم داده‌ها به چند زیرمجموعه کوچکتر مساوی
 - یک زیرمجموعه برای validation و بقیه برای آموزش
 - تغییر زیرمجموعه validation و تکرار
 - K-fold cross-validation

نکاتی در مورد انتخاب مجموعه validation برای overfitting جلوگیری از

cross-validation: random sub-sampling



مشکل: پیچیدگی محاسباتی بیشتر نسبت به داشتن یک مجموعه جدا برای validation (hold-out validation)

نکاتی در مورد انتخاب مجموعه validation برای overfitting جلوگیری از

□ نوع دیگر leave-p-out :cross-validation

■ استفاده از p مثال برای مجموعه validation و باقی برای آموزش
□ مشکل: exhaustive

e.g., for $p = 1$:

