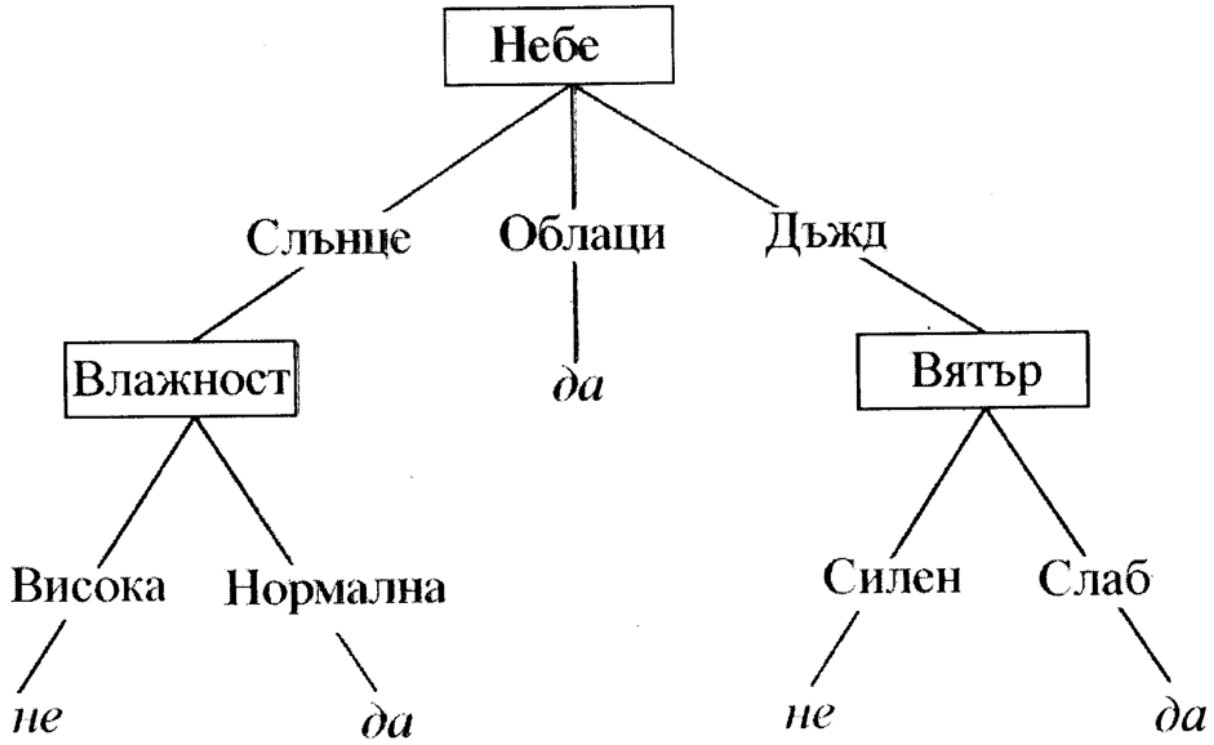
Класификационно дърво

- Всеки нетерминален възел – тест върху някой атрибут
- Всеки клон съответства на една от възможни стойности на проверявания атрибут
- Всеки терминален възел (листо) – на една от възможни стойности на целевия атрибут (класификация

Пример



Класификационното дърво на понятието *Игра-на-тенис*

$$(Небе = Слънце \wedge Влажност = Нормална)$$
$$\vee (Небе = Облаци)$$
$$\vee (Небе = Дъжд \vee Вятър = Слаб)$$

Подходящи за задачи:

- Примери – представени като конюнкция от ограничения: атрибут= стойност
- Целева функция – приема дискретно множество от значения
- Описанието на търсеното понятие може да е дизюнктивно
- Обучаващите данни могат да съдържат грешки
- Обучаващите данни могат да съдържат неизвестни (липсващи) стойности на атрибути

Класификационни задачи – съотнасяне на примери в една от възможни категории от едно предварително известно дискретно множество

Базов алгоритъм (условия за спиране)

ID3(Примери, Цел_атрибут, Атрибути)

Примери са обучаващите примери, *Цел_атрибут* е атрибутът, чиято стойност трябва да бъде предсказана, а *Атрибути* са останалите атрибути на примери, които се тестват от наученото класификационно дърво. Алгоритмът връща класификационното дърво, което коректно класифицира зададените *Примери.*

- **Създай** най-горен възел - *Корен* на дървото
- **Ако** всички *Примери* са положителни, **Върни,** като наученото, дърво с един единствен възел – *Корен*, маркиран със знака “+”.
- **Ако** всички *Примери* са отрицателни, **Върни,** като наученото, дърво с един единствен възел – *Корен*, маркиран със знака “-”.
- **Ако** *Атрибути* е празното множество, **Върни,** като наученото, дърво с един единствен възел – *Корен*, маркиран със знака, който съвпада с най-често срещано сред *Примери* значение на *Цел_атрибут*.
- **Иначе**

Базов алгоритъм (основен цикъл)

Започни

- *A* ← този атрибут от *Атрибути*, който най-добре класифицира *Примери*
- Класификационен атрибут на *Корен* ← *A*
- За всяка възможна стойност v_i на *A* направи:
  - **Добави** в дървото новия клон под *Корен,* съответстващ на теста $A = v_i$
  - Нека *Примери*($A=v_i$) са подмножество от *Примери*, които имат стойността v_i на атрибута *A*
  - **Ако** *Примери*(v_i) е празното множество
    - **То** добави под този нов клон листо, маркирано със знака, който съвпада с най-често срещано сред *Примери* значение на *Цел_атрибут*
    - **Иначе** добави под този нов клон под-дърво ***ID3(****Примери*($A=v_i$)*, Цел_атрибут, Атрибути – {A}))*

Край

- **Върни** *Корен*

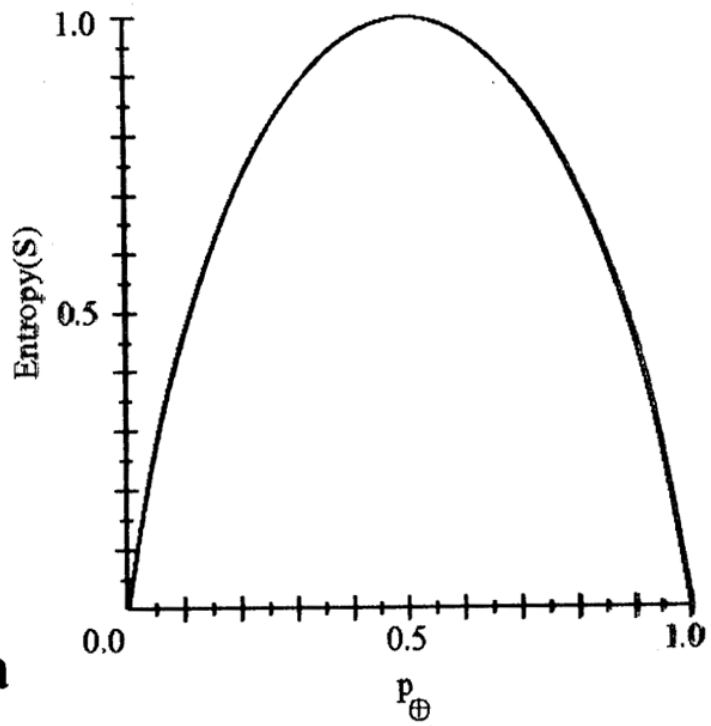
Кой атрибут е най-добрия класификатор?

Ентропия **- мярка за еднородността на примери:**

$$Entropy(S) \equiv -p_+ \log_2 p_+ - p_- \log_2 p_-$$

$p+$ е пропорцията на положителните примери в S

$p-$ е пропорцията на отрицателните примери в S.



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

Информационната печалба – мярка за очакваното намаляване на ентропията

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Стойности(A)} \frac{|S_v|}{|S|} Entropy(S_v) \quad S_v = \{s \in S \mid A(s) = v\}$$

Обучаващите примери на понятието

Игра-на-тенис

Ден	Небе	Температура	Влажност	Вятър	Игра-на-тенис
D1	Слънце	Горещо	Висока	Слаб	не
D2	Слънце	Горещо	Висока	Силен	не
D3	Облаци	Горещо	Висока	Слаб	да
D4	Дъжд	Топло	Висока	Слаб	да
D5	Дъжд	Студено	Нормална	Слаб	да
D6	Дъжд	Студено	Нормална	Силен	не
D7	Облаци	Студено	Нормална	Силен	да
D8	Слънце	Топло	Висока	Слаб	не
D9	Слънце	Студено	Нормална	Слаб	да
D10	Дъжд	Топло	Нормална	Слаб	да
D11	Слънце	Топло	Нормална	Силен	да
D12	Облаци	Топло	Висока	Силен	да
D13	Облаци	Горещо	Нормална	Слаб	да
D14	Дъжд	Топло	Висока	Силен	не

Entropy([9+,5-]) = -(9/14)log2(9/14) – (5/14)log2(5/14) = 0.940.

Илюстративен пример (1)

$Стойности(Вятър) = \{Слаб, Силен\}$

$S = [9+,5-]$

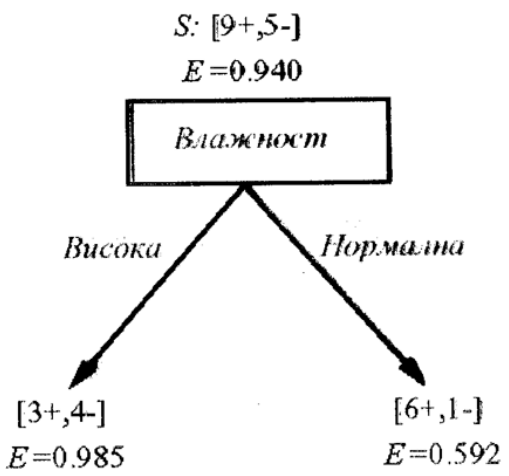
$S_{Слаб} = [6+,2-]$

$S_{Силен} = [3+,3-]$

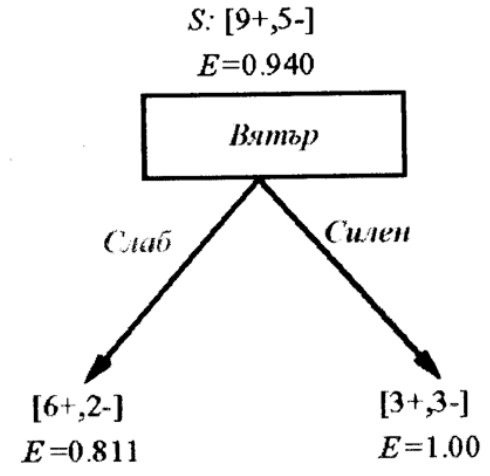
$$Gain(S, Вятър) = Entropy(S) - \sum_{v \in \{Слаб, Силен\}} \frac{|S_v|}{|S|} Entropy(S_v) =$$

$$= Entropy(S) - (8/14) Entropy(S_{Слаб}) - (6/14) Entropy(S_{Силен}) =$$

$$= 0.940 - (8/14)0.811 - (6/14)1.0 = 0.048$$



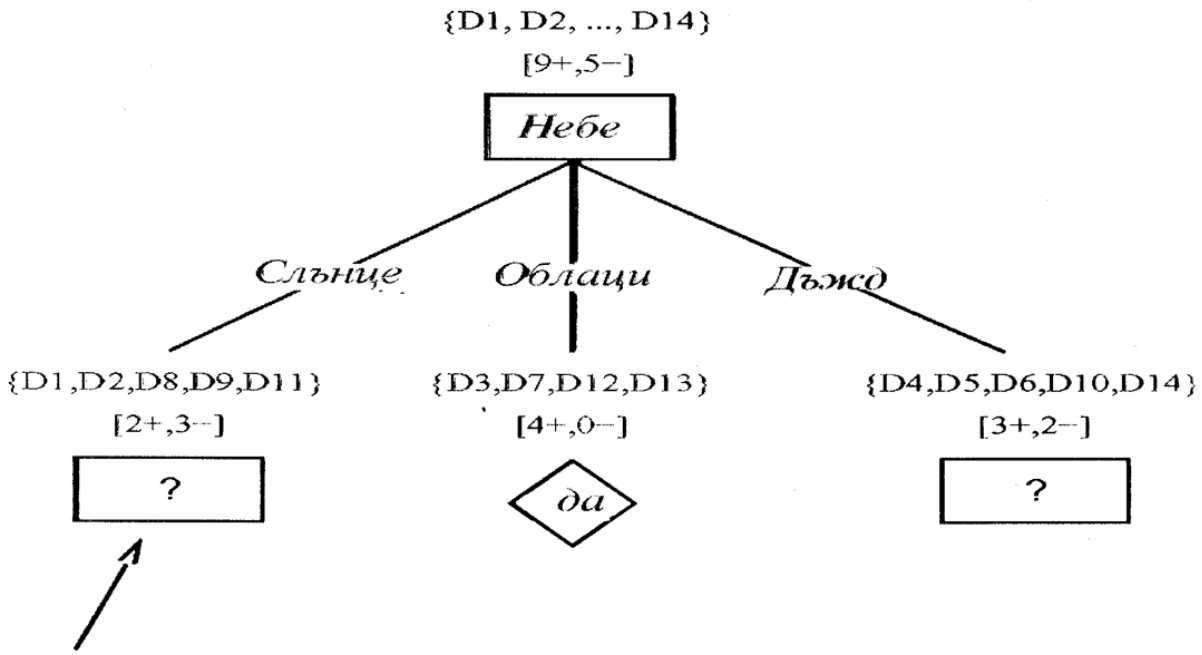
Gain (S, Влажност)
= .940 - (7/14).985 - (7/14).592
= .151



Gain (S, Вятър)
= .940 - (8/14).811 - (6/14)1.0
= .048

Gain(S, Небе) = 0.246; *Gain(S, Влажност) = 0.151;*
Gain(S, Вятър) = 0.048; *Gain(S, Температура) = 0.029*

Пример



Кой атрибут трябва да се тества тука?

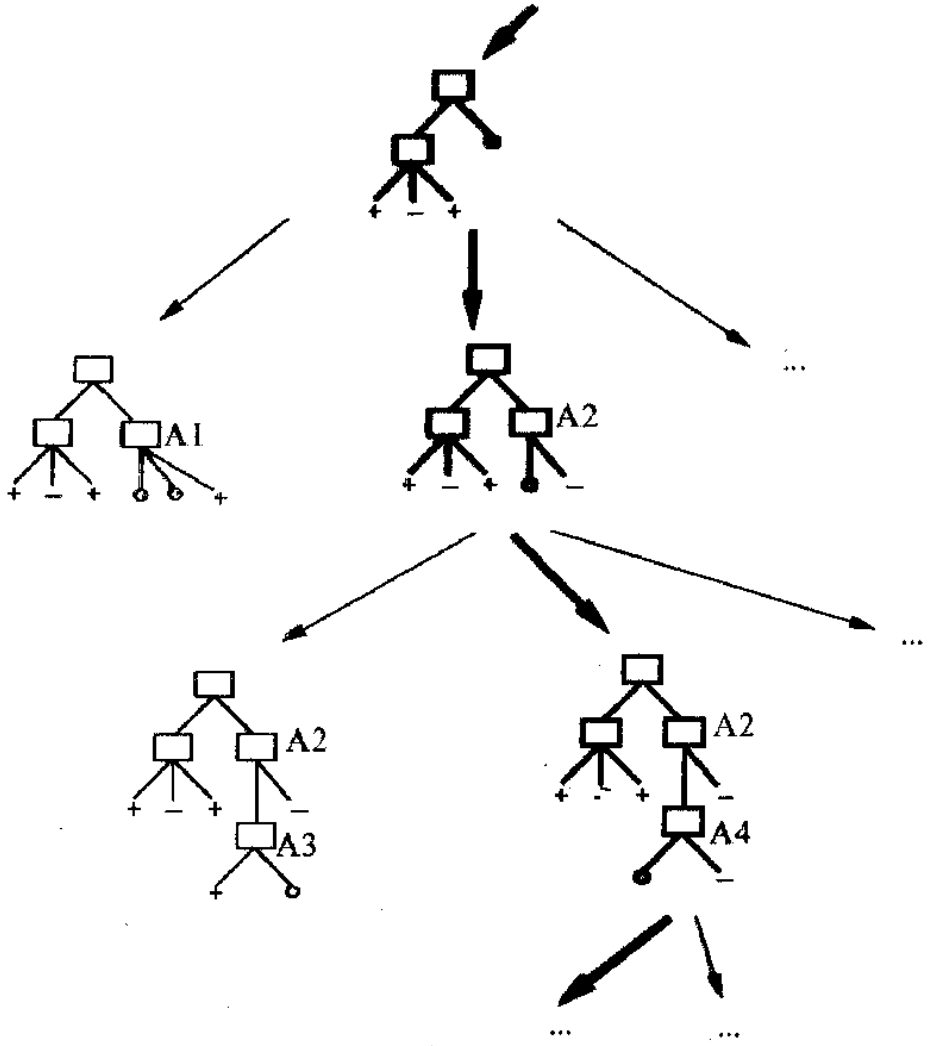
$S_{Слънце} = \{D1,D2,D8,D9,D11\}$

$Gain\ (S_{Слънце}, Влажност) = .970 - (3/5)\ 0.0 - (2/5)\ 0.0 = .970$

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

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

Търсене в пространство на класификационни дървета



Възможности и ограничения на ID3

- Пространството на хипотези на ID3 се състои от всички класификационни дървета и е *пълното* пространство на функции с дискретни стойности, относително наличните атрибути.
- При търсене в пространството от възможни класификационни дървета ID3 обработва само една единствена текуща хипотеза.
- В своята чиста форма ID3 *никога не се връща при търсенето*. След като веднъж избере някой атрибут за тестване на определеното ниво от дървото, алгоритмът никога не се връща да преразгледа този свой избор.
- На всяка стъпка от търсенето ID3 използва *всички* обучаващи примери, за да получи статистически обосновани решения за това, как да направи по-съвършена текущата хипотеза.

Индуктивното пристрастие на ID3

Късите дървета се предпочитат пред дългите.

Дърветата, които поместват атрибути с висока информационна печалба по-близо до корена си, се предпочитат пред тези, които не го правят

Обучаващи примери

Ден	Небе	Температура	Влажност	Вятър	Игра-на-тенис
D1	Слънце	Горещо	Висока	Слаб	не
D2	Слънце	Горещо	Висока	Силен	не
D3	Облаци	Горещо	Висока	Слаб	да
D4	Дъжд	Топло	Висока	Слаб	да
D5	Дъжд	Студено	Нормална	Слаб	да
D6	Дъжд	Студено	Нормална	Силен	не
D7	Облаци	Студено	Нормална	Силен	да
D8	Слънце	Топло	Висока	Слаб	не
D9	Слънце	Студено	Нормална	Слаб	да
D10	Дъжд	Топло	Нормална	Слаб	да
D11	Слънце	Топло	Нормална	Силен	да
D12	Облаци	Топло	Висока	Силен	да
D13	Облаци	Горещо	Нормална	Слаб	да
D14	Дъжд	Топло	Висока	Силен	не

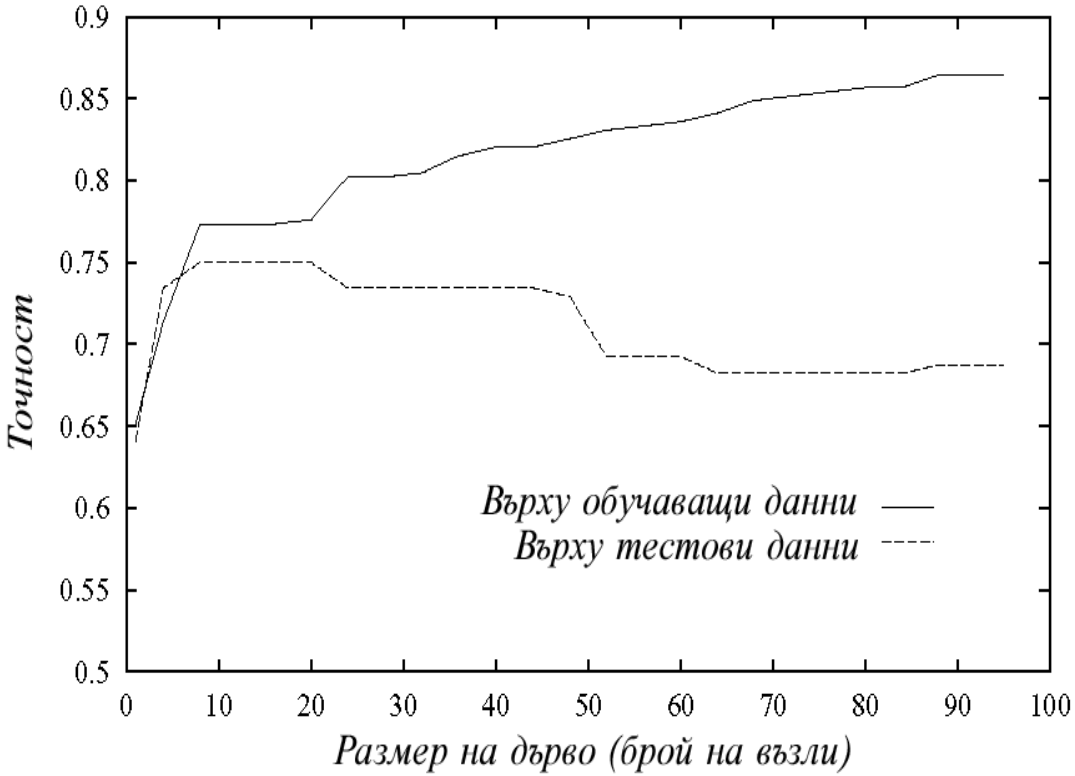
Класификационното дърво на понятието *Игра-на-тенис*

Нов (сгрешен) пример:

<*Небе=Слънце, Температура=Горещо,Влажност=Нормална,Вятър=Силен,Игра-на-тенис=не*>

Пренагаждане на хипотези

Определение. При зададено множество от хипотези H, ще казваме, че хипотезата $h \in H$ е *пренагодена* (overfitting) към обучаващите данни, ако съществува някоя алтернативна хипотеза $h' \in H$, такава че h има по-малка грешка от h' върху множеството от обучаващите примери, но h' има по-малка грешка от h върху цялото множество от примери.



Избягване на пренагаждането

- Подходи, спиращи изграждане на дървото преди то да стигне точката, където перфектно класифицира всички обучаващите примери.
- Подходи, позволяващи построяване на пренагоденото дърво, което след това се подлага на допълнително окастряне.

Какъв критерий трябва да се използва за определяне на размера на финалното дърво?

• Използване на отделното множество от примери, различаващо се от обучаващите примери.
• Използване за обучение на всички достъпни данни, но прилагане на статистическия тест за оценяване, дали разширяването (или изрязването) на конкретния възел може да доведе до подобрение в поведението на дървото върху данни извън обучаващото множество.
• Използване на определена мярка за кодиране на сложността на обучаващите примери и класификационното дърво, която спира разрастването на дървото, когато размерът на това кодиране е минимален.

Допълнително подрязване, минимизиращо грешката (Quinlan 1987)

- Разглеждането на всеки решаващ (нетерминален) възел в дървото като възможен кандидат за изрязване
- Изрязването на възела - изхвърлянето от дървото на цялото под-дърво с корен в дадения решаващ възел и замяната му с възел-листо, класификацията на който се определя от най-често срещан клас измежду обучаващите примери, асоциирани с възела
- Възлите се премахват само, ако полученото подрязано дърво има по-добро класификационно поведение от оригиналното (не подрязаното) дърво върху потвърждаващото множество.
- Възлите се изрязват итеративно, започвайки с възела, чието отстраняване води до най-голямото увеличение на точността на класификационното дърво върху потвърждаващото множество.

Пример

Класификационното дърво на понятието *Игра-на-тенис*

Първите два най-леви пътища в дървото се превръщат в следните правила:

АКО $(Небе = Слънце) \wedge (Влажност = Висока)$ *ТО* *(Игра-на-тенис = не)*

АКО $(Небе = Слънце) \wedge (Влажност = Нормална)$ *ТО* *(Игра-на-тенис = да)*

Обработка на непрекъснати атрибути

Създаване на един дискретен атрибут за проверка на непрекъснатият:

Температура = 24.5
(Температура >22) = T, F

Температура	5	9	15	21	25	30
Игра-на-тенис	Не	Не	Да	Да	Да	Не

Обработка на обучаващите примери с липсващи стойности на атрибути

Да предположим, че <*x, c(x)*> е един от обучаващите примери в *S* и че стойността *A(x)* е неизвестна.

Стратегии за работа с липсващите стойности на атрибути:

1. Заместване със стойността, която се среща най-често сред обучаващите примери от същия възел.
2. Заместване с най-често срещаната стойност на атрибута сред обучаващите примери в същия възел, които имат същата класификация *c(x)*.
3. На всяка възможна стойност на атрибута се присвоява определена вероятност.

Алтернативни мерки за избор на атрибути

Относителната печалба (gain ratio) (Quinlan 1986) - наказва атрибути с прекалено голям брой на възможни стойности чрез включване в себе си на елемент, наричан *информация за разделяне* (split information), който е чувствителен към това, на колко части и доколко еднообразно атрибутът разделя данните: когато

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

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

Проблеми: когато $|S_i| \approx |S|$ за някои от стойностите на атрибута.

Gini индекс

Classification and Regression Trees (CART) – алтернативен начин за научаване на *двоични* класификационни дървета (L. Briman др. 1984)

CART използва друга мярка за оценяване на еднородността/ неоднородността на едно множество от обучаващи примери:

$$Gini(S) = 1 - \sum_{i=1}^{k} p_i^2,$$

Където:

p_i - вероятността, че един случайно избран пример от S принадлежи към клас i.
k - броят на класове.

За оценка на тази вероятност може да се използва пропорцията на примери от този клас в S.

Двоично разделяне на номинален атрибут

Атрибутът *A* е номинален и има *v* различни стойности $\{a_1, a_2, \ldots, a_v\}$, които се срещат измежду обучаващите примери.

За определяне на най-доброто двоично разбиване на *A*, CART разглежда всички възможни подмножества на стойностите на атрибута.

Всяко подмножество S_A може да се разглежда като кандидат за двоичен тест на атрибута A във вида $A \in S_A$?

Един пример удовлетворява този тест, ако стойността на атрибута *A* в него присъства в списъка със стойностите, съдържащи се в S_A.

Ако *A* има *v* различни известни стойности, то броят на възможни подмножества е 2^v.

Пример: атрибут *Небе = {Слънце, Облаци, Дъжд}*.
Възможните подмножества: *{Слънце, Облаци, Дъжд}, {Слънце, Облаци}, {Слънце, Дъжд}, {Облаци, Дъжд}, {Слънце}, {Облаци}, {Дъжд}* и *{}*.
Премахваме от разглеждането *{Слънце, Облаци, Дъжд}* и *{}* - не дават никакво разбиване.
Следователно, има *2v – 2* възможни двоични разбивания на множеството примери *S* чрез двоичен тест върху *A*.

Двоично разбиване на непрекъснат атрибут

Ако атрибутът *A* има непрекъснати стойности, за избор на двоичното разбиване (прага *c*) се прилага същата стратегия, като и в C4.5 - стойностите на непрекъснатия атрибут се подреждат по големина и кандидатите за оптималният праг *c* се избират между съседните стойности на атрибута, за които съответните значения на целевия атрибут са различни.

Всеки кандидат праг разбива множеството от примери *S* на два подмножества:

$$S_+ = \{x \in S, A(x) \le c\} \qquad S_- = \{x \in S, A(x) > c\}$$

И в двата случая (т.е. когато атрибутът *A* е номинален или непрекъснат) *Gini* индекс $Gini_A(S)$ за всеки двоичен тест се определя като претеглената сума на *Gini* индекси на всяко от двете подмножества, получени при разделяне на *S:*

$$Gini_A(S) = \frac{|S_+|}{|S|} Gini(S_+) + \frac{|S_-|}{|S|} Gini(S\ ), \textit{където}$$

$S_+ \in S$ - е множество примери, удовлетворяващи тест $A \in S_A$?

$S_- \in S$ - множество примери, които не удовлетворяват този тест

Най-доброто двоично разбиване на множеството примери *S* - тест върху *A*, който води до най-голямото намаление на неоднородността на множеството, измервано чрез *Gini* индекс:

$$\Delta Gini(S, A) = Gini(S) - Gini_A(S)$$