Classificació: arbres de decisió

Ramon Sangüesa i Solé

P03/05054/01035

UOC
www.uoc.edu

Índex

Introducció

Els arbres de decisió són un dels models de classificació més utilitzats. La seva interpretació prou natural, la seva relació amb tècniques de classificació establertes i provades, en particular, i també els diversos mètodes per a traduir els models resultats en col·leccions de regles de classificació, els fan versàtils i aplicables. 

Objectius

L'estudi dels materials associats a aquesta assignatura permetran a l'estudiant d'assolir els objectius següents:

1. Conèixer l'origen i les idees fonamentals que hi ha darrere els arbres de decisió.

2. Saber els detalls dels mètodes de construcció més freqüents i utilitzats.

1. Introducció: l'estructura dels arbres de decisió

La millor manera de començar és amb l'arbre de decisió que resulta d'aplicar un mètode força conegut, el mètode ID3, a un conjunt de dades sobre els clients del gimnàs.

El conjunt de dades sobre els clients del gimnàs consisteix en una sèrie de registres que apleguen els valors de les dades següents:

- Identificador del client.

- Nivell físic del client: aquest atribut ens indica, segons els resultats de les proves físiques a les quals ha estat sotmès el client en inscriure's al gimnàs, si el seu estat físic és prou bo. Els valors que pren aquest atribut són 'Baix', 'Normal' i 'Alt', que corresponen a un estat físic cada cop millor.

- Intensitat de l'activitat 1: fa referència al nivell d'esforç que correspon a l'activitat esportiva principal que desenvolupa el client. Aquest atribut pren valors en les categories 'Baixa', 'Mitjana' i 'Alta'.

- Intensitat de l'activitat 2: aquest atribut indica el mateix que l'atribut anterior, però referit a la segona activitat que desenvolupa el client.

- Classe: atribut que pren els valors 0 o 1 i que indica si el client efectua una combinació correcta de nivell i activitat o no. Aquesta assignació a cada classe l'ha efectuada manualment un especialista en preparació esportiva.

Client	Nivell físic	Intensitat de l'activitat 1	Intensitat de l'activitat 2	Classe
1	Normal	Mitjana	Baixa	0
2	Baix	Baixa	Baixa	1
3	Normal	Alta	Alta	1
4	Alt	Alta	Alta	1
5	Alt	Baixa	Baixa	0
6	Baix	Mitjana	Mitjana	1
7	Normal	Mitjana	Mitjana	0
8	Alt	Alta	Alta	1

El problema que es plantegen els mètodes de construcció d'arbres de decisió és el següent: quina és la millor seqüència de preguntes per a aconseguir saber, a partir de la descripció d'un objecte en termes dels seus atributs, a quina classe correspon? Evidentment, "la millor seqüència" es pot entendre com la que, amb el mínim nombre de preguntes, torni una resposta prou acurada.

En altres paraules, la "millor seqüència" és la que efectua les preguntes més discriminants, aquelles les respostes de les quals permeten de descartar un nombre d'objectes diferents del que considerem més ampli i, per tant, d'arribar més ràpidament a decidir a quina classe pertany.

Exemple de millor seqüència de preguntes

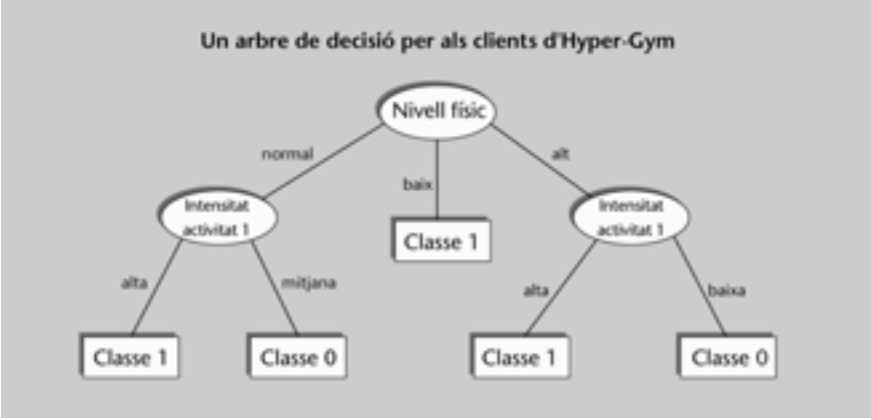
Per a establir, per exemple, si un client pertany a la classe 0 o a la classe 1, què és millor?: preguntar primer pel seu estat físic o per la intensitat de la primera activitat que desenvolupa?

Què té a veure això amb un arbre? Doncs que la manera d'organitzar les preguntes porta amb prou naturalitat a una estructura arbòria. En efecte, un arbre de decisió està format per nodes (nodes de decisió) que efectuen una pregunta sobre el valor d'un atribut. Les diferents respostes que s'hi poden donar generen altres nodes de decisió. D'aquesta manera, es va construint l'arbre.

L'interès d'aquests models és que, a més, permeten una traducció prou natural a llistes de regles de decisió.

Exemple de construcció d'un arbre de decisió

En el cas dels nivells físics i intensitats d'exercici d'un client d'Hyper-Gym, podem obtenir l'estructura següent:



N'hi ha prou de resseguir el camí que va dins l'arbre des de l'arrel fins a cadascuna de les fulles per a anar construint una conjunció de condicions sobre els valors dels atributs (allò que es pregunta a cada node) i d'aquesta manera construir l'antecedent d'una regla de decisió. Aquest arbre es pot traduir al conjunt de regles següent:

Si (*Nivell_físic* = 'Normal') & (*Intensitat_activitat_1* = 'Mitjana') aleshores *classe* = 0
Si (*Nivell_físic* = 'Normal') & (*intensitat_activitat_1* = 'Alta') aleshores *classe* = 1
Si (*Nivell_físic* = 'Baix') aleshores *classe* = 1
Si (*Nivell_físic* = 'Alt') & (*intensitat_activitat_1* = 'Alta') aleshores *classe* = 1
Si (*Nivell_físic* = 'Alt') & (*intensitat_activitat_1* = 'Baixa') aleshores *classe* = 0

Els mètodes de construcció d'arbres de decisió comencen per intentar obtenir una condició, una prova sobre els valors d'un atribut, tal que efectuï la millor partició sobre els conjunts de dades; és a dir, que separi els diversos casos en grups tan diferents entre si com sigui possible, però de manera que els casos que hi ha dins de cada partició tinguin la màxima semblança possible entre si, ja que aquest és l'objectiu d'un procediment de classificació. S'aconsegueix la màxima homogeneïtat quan tots els casos que apareixen en una partició pertanyen a la mateixa classe.

Per tant, l'important en els mètodes de construcció d'arbres de decisió és escollir quin és l'atribut que separa millor els casos existents.

Amb petites variacions segons els diversos mètodes, el procediment de construcció de l'arbre procedeix iterativament. Així, si el primer node que es considera és un atribut que té tres valors possibles (per exemple, 'Alt', 'Baix' i 'Normal'), llavors el conjunt de dades es parteix en tres subconjunts que corresponen als casos per als quals l'atribut corresponent té el valor 'Alt', el valor 'Normal' i el valor 'Baix'. La pregunta pel valor 'Alt' es converteix en un node de decisió, i també les preguntes o condicions sobre els valors 'Baix' i 'Normal'. En aquest moment, l'arbre ja té un nivell.

A les particions de les dades que resulten s'hi torna a aplicar el procediment fins que es compleix una condició de final, que és diferent segons cada mètode, però que en general podem dir que intenta obtenir particions on tots els casos que li corresponen pertanyen a la mateixa classe, o on la majoria dels casos són de la mateixa classe.

Per exemple, si ens fixem en aquesta relació entre els valors dels atributs i els casos dins la base de dades en què efectivament es compleix això, podem veure que en el primer nivell de l'arbre dels clients i intensitats d'exercici tenim aquesta primera partició:

Condició	Casos (Clients)	Classe a què pertany
Nivell físic = 'Normal'	1	0
	3	1
	7	0
Nivell físic = 'Baix'	2	1
	6	1
Nivell físic = 'Alt'	4	1
	5	0
	8	1

Es pot observar que en aquest nivell el valor 'Baix' de l'atribut *Nivell físic* és suficient per a agrupar correctament dins la mateixa partició dos casos que per-

tanyen a la mateixa classe; és a dir, que dóna lloc a una partició (un subconjunt de les dades originals) completament homogènia perquè tots els seus casos pertanyen a la classe 1.

D'aquesta partició es pot deduir que el valor 'Baix' d'aquest atribut discrimina prou bé, però els altres dos ('Normal' i 'Alt') no tant, ja que les particions corresponents apleguen casos tant de la classe 0 com de la 1. Cal utilitzar els altres atributs per a veure si podem obtenir particions igualment homogènies.

Fixem-nos en les condicions que corresponen a les preguntes sobre els valors de l'atribut *Intensitat de l'activitat 1*. Posem aquí les combinacions de valors corresponents als camins possibles dins l'arbre:

Condició	**Casos (Clients)**	**Classe a què pertany**
Nivell físic = 'Normal' & *Intensitat de l'activitat 1* = 'Mitjana'	1	0
	7	0
Nivell físic = 'Normal' & *Intensitat de l'activitat 1* = 'Alta'	3	1
Nivell físic = 'Alt' & *Intensitat de l'activitat 1* = 'Alta'	4	1
	8	1
Nivell físic = 'Alt' & *Intensitat de l'activitat 1* = 'Baixa'	5	0

En aquest cas es pot veure que totes les particions que indueixen les diverses condicions són completament homogènies. Tots els casos que tenen un nivell físic normal i una primera activitat amb intensitat mitjana pertanyen a la classe 1. Tots els casos amb nivell físic normal i primera activitat amb intensitat alta pertanyen a la classe 0. I així successivament.

En la figura següent podem veure que a cada branca li corresponen els diversos casos:

En realitat, doncs, els algorismes de construcció d'arbres de decisió efectuen una partició de l'espai determinada pels atributs que es considerin. En el cas

de l'exemple, en què es consideren, obviant l'identificador de client, tres atributs (*Nivell físic*, *Intensitat de l'activitat 1* i *Intensitat de l'activitat 2*), aquests determinen regions en l'espai de tres dimensions. La resposta que donen les cadenes de preguntes que estan representades per les diverses regles és a quina subregió d'aquest espai cal ubicar un cas nou.

Activitat

1.1. Mediteu un moment per què en l'exemple que acabem de donar no cal preguntar per la intensitat de la segona activitat física desenvolupada pel client.

La clau de la qüestió per a cada mètode és com s'ha de trobar, sobre quins atributs cal preguntar i en quina seqüència per a obtenir arbres que tinguin la màxima capacitat predictiva i alhora compleixin altres paràmetres de qualitat (per exemple, que no tinguin massa nivells).

En altres paraules, la clau rau en la manera en què aconseguim que a cada pas s'obtingui allò que hem anomenat *la millor partició*.

Els diversos mètodes que s'empren per a assolir aquest objectiu es distingeixen per les característiques següents:

- Capacitat per a utilitzar atributs amb més de dos valors.
- Ús de diferents mesures d'homogeneïtat de les classes.
- Capacitat per a preguntar a cada node sobre més d'un sol atribut.

A continuació revisarem els mètodes més coneguts: ID3, C4.5, CART, CHAID i LMDT.

2. Mètodes de construcció d'arbres de decisió per a classificació: ID3 i C4.5

El mètode ID3 es deu a J. Ross Quinlan, un investigador australià, que el va proposar el 1986 (Quinlan, 1986). L'acrònim que dóna nom al mètode correspon a l'expressió anglesa *iterative dichotomizer 3*, és a dir: *dicotomitzador iteratiu 3*. En altres paraules, és un procés que repeteix un "tall en dos"* fins que es compleix una determinada condició.

Introduirem una mica de nomenclatura i notació:

Suposem que disposem d'un conjunt de m casos. Designem aquest conjunt de casos per $X = \{x_j\}_{j=1,m}$. Cal recordar que cada x_j és, en realitat, una tupla definit respecte al conjunt d'atributs de partida.

- El **conjunt d'atributs de partida** és $A = \{A_k\}_{k=1,n}$. En el cas de l'exemple, tenim el següent:

  A_1 = *Nivell físic*,
  A_2 = *Intensitat de l'activitat 1*,
  A_3 = *Intensitat de l'activitat 2*.

- El **conjunt de casos** és $X = \{x_j\}_{j=1,8}$, on, per exemple, x_3 = {'Normal', 'Alta', 'Alta'}.

- El **conjunt de valors** d'un atribut, A_i, es denota per $V(A_i)$. Per exemple, $V(A_1)$ = {'Alt', 'Normal', 'Baix'}. El valor que pren un A_i per a un cas x_j es denota per $A_i(x_j)$. Per exemple, el valor de l'atribut 3 per al cas 2 és $A_3(x_2)$ = = Baixa.

- El conjunt de casos que prenen un determinat valor v per a un atribut A_i determinat es denota per $A_i^{-1}(v)$.

$$A_i^{-1}(v) = \{x \in X: A_i(x) = v\}.$$

Per exemple, el conjunt de casos amb nivell físic baix correspon als casos en què el primer atribut és igual a 'Baix'. Formalment, amb aquesta notació:

$$A_i^{-1}(\text{'Baix'}) = \{2, 6\}.$$

- El conjunt de classes es denota per $C = \{C_i\}_{i=1,k}$. En el nostre exemple, $k = 2$.

El **mètode ID3** tracta de trobar una partició que asseguri la màxima capacitat predictiva i la màxima homogeneïtat de les classes.

2.1. Mesures d'homogeneïtat

Per a mesurar l'homogeneïtat de cada subconjunt hi ha diverses mesures. Cerquen, d'una manera o d'una altra, assignar valors extrems al cas en què la partició que es consideri contingui només casos d'una sola classe. També ens permeten de comparar particions no completament uniformes, tenint en compte que la seva diversitat interna queda reflectida de manera numèrica i això forneix una base per a la comparació.

A continuació comentarem les diferents mesures de desordre de què hem parlat. De fet, es tracta de veure quin grau de varietat o de desordre hi ha en la partició que resulta de fixar un valor v determinat d'entre els diversos valors que pot prendre un atribut A_i.

Exemple de mesura de desordre

Podem escollir l'atribut *Nivell físic*. Veiem que el valor 'Baix' genera una partició que és més homogènia que la que genera el valor 'Alt' o la que genera el valor 'Normal':

Valor de l'atribut *Nivell físic*	Casos (*Clients*)	Classe a què pertany
Nivell físic = 'Normal'	1	0
	3	1
	7	0
***Nivell físic* = 'Baix'**	**2**	**1**
	6	**1**
Nivell físic = 'Alt'	4	1
	5	0
	8	1

Si fixem la proporció de casos de la classe 0 de la manera següent:

$$\frac{\text{Nombre de casos de clase 0}}{\text{Nombre de casos de la partició}},$$

tenim que cada partició presenta les proporcions següents, com a mesures primitives de diversitat de cada una d'aquelles:

Partició	Valor de l'atribut *Nivell físic*	Casos	Proporció
1	Normal	{1, 3, 7}	2/3
2	Baix	{2, 6}	0
3	Alt	{4, 5, 8}	1/3

Sota aquesta mesura de desordre, la partició 2, que correspon als casos en què el nivell físic és baix, té la màxima homogeneïtat (valor de proporció = 0). Les altres dues particions encara no són prou homogènies, però sota el criteri que hem fet servir, la partició 1 sembla més desordenada o menys homogènia que la partició 2 (2/3 > 1/3). Quin és el comportament dels altres atributs? El veiem en les taules que hi ha a continuació:

Valor de l'atribut *Intensitat de l'activitat 1*	**Casos (*Clients*)**	**Classe a què pertany**	**%**
Baixa	2	1	50
	5	0	
Mitjana	1	0	66
	6	1	
	7	0	
Alta	3	1	0
	4	1	
	8	1	

Valor de l'atribut *Intensitat de l'activitat 2*	**Casos (*Clients*)**	**Classe a què pertany**	**%**
Baixa	1	0	66
	2	1	
	5	0	
Mitjana	6	1	50
	7	0	
Alta	3	1	0
	4	1	
	8	1	

La proporció que hem proposat en aquest exemple no és més que una mesura orientativa a tall d'explicació.

Normalment s'assigna a aquest cas el valor 0, i com més heterogènia sigui la partició –és a dir, com més casos procedents de classes diferents tingui–, llavors la mesura adquireix un valor més gran. D'alguna manera, intenta mesurar el "desordre" que hi ha dins de cada partició.

La mesura del desordre típica és l'**entropia**, que és una propietat de les distribucions de probabilitat dels diversos atributs i que ara comentem amb més detall.

La **mesura del desordre** és una "mesura d'informació". Efectivament, la informació s'ha associat amb el grau de "sorpresa" que aporta un senyal. Si esperem que una dada tingui un valor determinat, si per exemple, sabem que normalment la renda d'un client és de 3,5 milions i apareix un client amb 30 milions de renda, llavors hem de considerar aquest fet com a "sorprenent"; és a dir, és un valor més informatiu que d'altres de més propers als 3 milions.

La propietat anterior s'expressa relacionant els valors que pot prendre una variable amb la probabilitat d'aparició que té; és a dir, amb la **distribució de probabilitat d'una variable**.

El desordre es mesura en bits. Si tenim una variable X que pot prendre els valors x_1, ..., x_n segons una distribució $P(X)$, llavors el grau de sorpresa de cada valor ha d'estar en relació amb la probabilitat d'aparició d'aquest valor. En principi, valors més baixos de probabilitat haurien de ser més sorprenents.

La manera de trobar els bits d'informació continguts en la distribució de probabilitat d'un atribut està determinada per l'expressió que donem a continuació:

$$I(X) = -\sum_{1}^{n} \log_2 p(x_i)$$

Tenint en compte que els logaritmes per a valors reals entre 0 i 1 són negatius, la suma total és positiva.

Què té a veure el desordre amb l'homogeneïtat d'una classe? Fixeu-vos que aquesta propietat està en relació amb un atribut que prenem com el decisiu per qualificar una classe d'homogènia. Hem d'intentar obtenir una classe homogènia; per tant, on predomini un dels valors de l'atribut. En conseqüència, necessitem una mesura que reflecteixi el fet que, si tots els valors són el mateix, la qualitat és màxima; això és precisament el que fa la mesura d'informació.

El guany d'informació.

De fet, a cada pas de la construcció d'un arbre no ens interessa l'atribut que aporta més informació, sinó el que ens dóna la màxima diferència d'informació respecte a la partició actual si l'efectuem tenint en compte els seus valors. És a dir, ens interessa el màxim **guany d'informació**. Es tracta, doncs, d'escollir a cada pas l'atribut que maximitzi el guany d'informació.

Donada una partició, el **guany d'informació** es troba fent l'aproximació que la probabilitat de cada valor és la seva freqüència d'aparició observada.

El **guany d'informació** queda definit de la manera següent:

$$G(X,A_k) = I(X,C) - E(X,A_k),$$

on cada element s'explica a continuació:

a) $I(X,C)$ és la informació associada a les particions induïdes per un conjunt de classes donat, C, respecte al conjunt de casos X. Es defineix de la manera següent:

$$I(X,C) = -\sum_{C_i \in C} p(X,c_i)\log_2 p(X,c_i)$$

on $p(X,c_i)$ és la probabilitat que un exemple pertanyi a una classe donada c_i. S'acostuma a fer l'aproximació que aquesta probabilitat és la freqüència d'aparició observada de casos que pertanyen a la classe c_i, que és l'estimador d'aquesta probabilitat:

$$p(X,c_i) = \frac{\#c_i}{\#X} \ .$$

En aquesta expressió, $\#c_i$ és el nombre de casos del conjunt de casos X que pertanyen a la classe c_i, i $\#X$ és el nombre de casos total del conjunt de casos.

Exemple d'obtenció

En l'exemple que hem anat seguint al llarg d'aquest apartat sobre el nivell físic dels clients de la cadena de gimnasos Hyper-Gym, el nombre de casos seria $\#X = 8$. Aleshores, la probabilitat que un cas pertanyi a la classe 0 és la següent:

$$P(X,C_0) = \frac{3}{8}.$$

b) $E(X,A_k)$ és la informació esperada de l'atribut A_k pel que fa al conjunt de casos X. És a dir, indica el nivell de diversitat d'aquest atribut (els diversos valors que pot prendre) en el conjunt de casos X.

$E(X,A_k)$ és una mesura de la diversitat que introdueix en les particions el fet d'escollir l'atribut A_k. S'expressa de la manera que veiem a continuació:

$$E(X,\ A_k) = \sum_{V_i \in V^{(Ak)}} p(X,\ A_k^{-1}(v_i)) \cdot I(A_k^{-1},\ C)\ ,$$

on $p(X,\ A_k^{-1}(V_i))$ és la probabilitat que un cas tingui el valor v_i en l'atribut A_k. Correspon a fer l'aproximació a partir de les freqüències observades següents:

$$p(X,\ A_k^{-1}(v_i)) = \frac{\#(A_k^{-1}(v_i))}{\#\mathrm{X}}.$$

És a dir, correspon al nombre de casos que mostren el valor v_i en l'atribut A_k respecte al total de casos existents.

El **guany d'informació** mesura en quin grau un atribut determinat fa augmentar o disminuir el desordre de les particions que hi pot haver a partir de comparar el desordre que induiria respecte al que hi ha en un moment determinat. La funció $I(X,C)$ mesura la diversitat de valors respecte a les classes existents.

La mesura de guany d'informació té un problema: els atributs que prenen molts valors diferents. Entre els atributs X_1, que pren els valors {'Sí', 'No'}, i X_2, que pren els valors {'Alt', 'Baix', 'Normal'}, normalment preferirà el segon. Per què? Ho veurem tot seguit.

Suposem el cas d'un atribut que tingui un valor diferent per a cada observació del conjunt de dades. En una situació com aquesta, quina seria la seva informació?

$$I(X) = -\sum_{1}^{n} \log_2 p(X_i) = 0\,,$$

perquè cada partició que generéssim seria completament homogènia. Per tant, seria l'atribut el que ens donaria sempre el màxim guany respecte al darrer test efectuat.

Lectura recomanada

Trobareu més informació amb relació a altres mètodes que empren les mesures d'homogeneïtat i les repercussions que en comporta la tria en l'article següent:

R. López de Màntaras (1991). "A Distance-Based Attribute Selection Measure for Decision Tree Induction". *Machine Learning* (vol. 6, núm. 1, pàg. 81-92).

Una mesura que intenta compensar aquesta tendència és la **mesura de raó del guany d'informació** (López de Màntaras, 1991). Aquesta mesura compensa l'efecte esmentat tenint en compte el nombre de possibles particions que generaria un atribut i la seva grandària respectiva (sense tenir en compte cap informació sobre la classe).

Les mesures comentades es fan servir per a altres mètodes i l'elecció d'una o d'una altra té repercussions importants quant als tipus d'atributs que s'escull i la forma final de l'arbre que s'obté.

2.2. Formalització de l'algorisme ID3

Formalitzem a continuació l'algorisme ID3 de construcció d'arbres de decisió, seguint la notació que hem proposat anteriorment. El presentem en forma de funció recursiva. La funció ID3 pren com a paràmetres d'entrada dos conjunts, X i A. El primer correspon al conjunt de casos, i el segon, al conjunt d'atributs. Mantenim la notació que hem presentat al principi. La funció crear_arbre(A_k) crea un arbre que té com a node arrel l'atribut A_k. Quan aquest atribut correspon a l'etiqueta d'una classe, el denotem per crear_arbre(C_i).

Farem una traça sobre un petit exemple per tal de veure com es va construint progressivament l'arbre a partir d'un conjunt de dades minúscul i fictici. Aquí el tenim:

Funció ID3 (**ent** X, A **són** conjunts) **retorna** arbre_de_decisió
var *arbre1*, *arbre2* **són** arbre_de_decisió;
A_{max} **és** atribut

opció
cas: ($\exists C_i \forall x_j \in X \Rightarrow x_j \in C_i$) **fer**
{Tots els casos són d'una mateixa classe}
arbre1 := crear_arbre(C_i)
cas: **no**($\exists C_i \forall x_j \in X \Rightarrow x_j \in C_i$) **fer**
opció:
cas $A \neq \emptyset$ **fer** {Hi ha algun atribut per a tractar}
A_{max}:= $max_{A_k \in A}\{G(X,A_k\}$
{Escollir l'atribut A_{max} que maximitzi el guany}
arbre1 := crear_arbre(A_{max}) {Crear un arbre amb arrel A_{max}}
Per a tot $v \in V(A_{max})$ **fer**
arbre2 := ID3($A_{max}^{-1}(v)$, $A - \{A_{max}\}$);
{Aplicar ID3 al conjunt de casos amb valor *v* a
l'atribut A_{max}, però sense considerar aquest atribut}
arbre1 := afegir_branca(*arbre1*,*arbre2*,*v*);
{connectar l'arbre resultant}
fper
cas A = $\emptyset$ **fer**
arbre1 := crear-arbre(classe_majoritària(*X*))
fopció
retorna *arbre1*
ffunció

Cas	Horari	Sexe	Nivell físic	Classe
1	Matí	Home	Alt	0
2	Matí	Dona	Normal	0
3	Migdia	Dona	Normal	1
4	Tarda	Dona	Normal	1
5	Tarda	Dona	Alt	0
6	Migdia	Dona	Baix	1
7	Tarda	Home	Baix	1
8	Matí	Dona	Normal	0

1) Comencem el procés. Consisteix en els càlculs següents:

a) Mesurem, en primer lloc, la informació continguda en el conjunt inicial de dades:

$$I(X, C) = -\frac{1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right) = 1 .$$

El primer terme correspon als casos de la classe 0 (1, 2, 5 i 8), i el segon, als de la classe 1 (3, 4, 6 i 7).

b) Mesurem l'entropia aportada per cadascun dels atributs:

- $E(X, Sexe) = \frac{2}{8}\left(-\frac{1}{2}\log_2\frac{1}{2}-\frac{1}{2}\log_2\frac{1}{2}\right)+\frac{6}{8}\left(-\frac{3}{6}\log_2\frac{3}{6}-\frac{3}{6}\log_2\frac{3}{6}\right) = 1$
- $E(X, Horari) = \frac{3}{8}(-1\log_2 1 - 0\log_2 0)+\frac{2}{8}(-0\log_2 0 - 1\log_2 1)+$

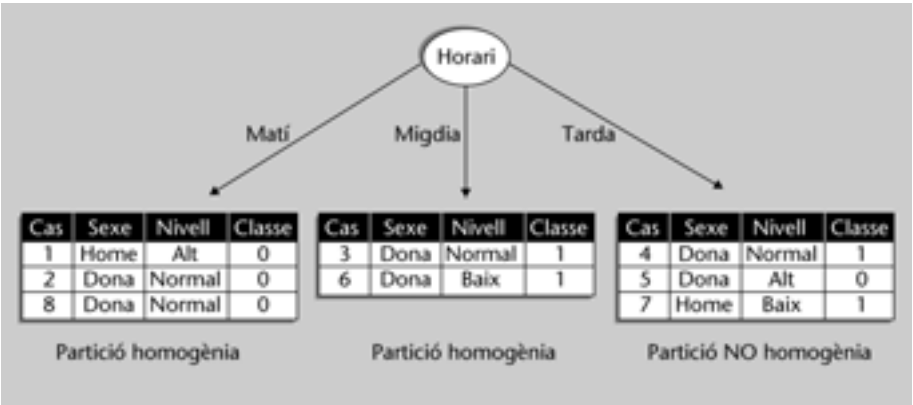
  $+\frac{3}{8}\left(-\frac{1}{3}\log_2\frac{1}{3}-\frac{2}{3}\log_2\frac{2}{3}\right)= 0{,}344$
- $E(X, Nivell\ físic) = \frac{2}{8}(-1\log_2 1 - 0\log_2 0)+\frac{4}{8}\left(-\frac{1}{2}\log_2\frac{1}{2}-\frac{1}{2}\log_2\frac{1}{2}\right)+$

  $+\frac{2}{8}(-0\log_2 0 - 1\log_2 1) = 0{,}5$

En la taula que veiem a continuació tenim el guany per a cadascun dels atributs:

Atribut	$I(X,C) - E(X,A_k)$
Horari	1 – 0,344 = 0,656
Sexe	1 – 1 = 0
Nivell físic	1 – 0,5 = 0,5

En conseqüència, l'atribut que aporta un canvi més important cap a la consecució de particions més homogènies és *Horari*. Per tant, la construcció del primer nivell de l'arbre correspon a l'esquema següent:



Com es pot veure, no totes les particions són homogènies; per tant, cal iterar el procés amb els atributs que encara no s'han tractat.

Primerament hem de trobar quin dels atributs {*Sexe*, *Nivell*} és el que té més guany.

2) Iterem l'aplicació de les fórmules que ja coneixem:

a) Calculem la informació inicial de la partició que interessa:

$$I(X, C) = -\frac{1}{3}\log_2\left(\frac{1}{3}\right)-\frac{2}{3}\log_2\frac{2}{3} = 0{,}918$$

El primer terme correspon al cas 5, i el segon, als casos 4 i 7.

b) Calculem ara l'entropia que aporta cada atribut:

- $E(X, Sexe) = \left(\frac{1}{3} - 1\log_2 1 - 1\log_2 1\right) + \frac{2}{3}\left(-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}\right) = \frac{2}{3}.$

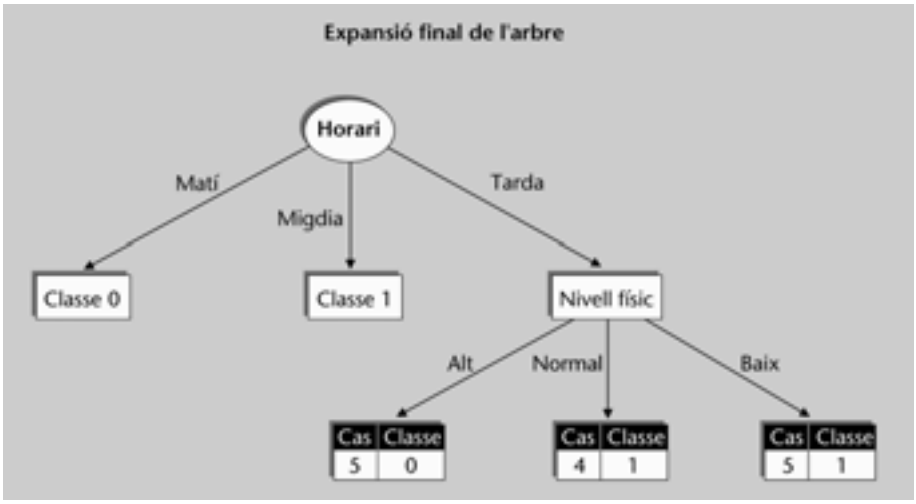
- $E(X, Nivell) = \frac{1}{3}\left(-0\log_2 0 - 1\log_2 1\right) + \frac{1}{3}\left(-1\log_2 1 - 0\log_2 0\right) +$

  $+ \frac{1}{3}\left(-0\log_2 0 - 1\log_2 1\right) = 0.$

c) Finalment, calculem el guany corresponent:

Atribut	$I(X,C) - E(X,A_k)$
Sexe	0,918 – 0,666 = 0,252
Nivell físic	0,918 – 0 = 0,918

3) Desenvolupem l'arbre de la manera corresponent:



En aquest moment totes les fulles de líarbre corresponen a particions homogènies. En el cas del subarbre de més a la dreta, les particions són trivialment homogènies, ja que contenen un únic cas, que forçosament no pot pertanyer més que a una única classe.

Horari = 'Matí' => Classe 0
Horari = 'Migdia' => Classe 1
Horari = 'Tarda & *Nivell físic*' = 'Alt' => Classe 0
Horari = 'Tarda & *Nivell físic*' = 'Normal' => Classe 1
Horari = 'Tarda' & '*Nivell físic*' = 'Baix' => Classe 1

L´exemple que acabem de veure ès molt senzill, peró ha posat en evidència uns quants aspectes dels métodes de construcció díarbres de decisió:

a) Els arbres de decisió fan una partició d'un espai amb tantes dimensions com atributs es fan servir per a descriure els casos.

b) Troben condicions lògiques per a descriure les característiques dels casos que cauen dins de cada partició.

c) Utilitzen mesures que avaluen el desordre de cada partició aportada per cada atribut en cada pas de l'algorisme.

Vegeu les regles d'associació en el mòdul "Regles d'associació" d'aquesta assignatura.

d) Es poden traduir fàcilment a regles.

Fixem-nos que aquest mateix exemple ja ens assenyala algun dels problemes comuns a altres mètodes de classificació que poden aportar els arbres: les classes d'un cas únic potser són massa específiques i no serveixen per a generalitzar. També és cert que aquest petit exemple és massa reduït per a poder aconseguir generalitzacions significatives.

L'ID3 també té una sèrie de problemes que li són propis. Bàsicament, genera massa branques a l'arbre. Això provoca el següent:

- Que les regles siguin clarament sobreespecialitzades.
- Que es redueixi innecessàriament el nombre de casos que corresponen a cada node, amb la qual cosa la informació que se n'extreu té cada cop menys suport estadístic perquè correspon a una mostra més i més petita.

El fet que es generin arbres amb aquestes característiques afecta també la capacitat predictiva del model resultant.

En el nostre exemple, considerem quina és la capacitat predictiva de cada branca de l'arbre.

Recordem les regles que han derivat de l'arbre. Les presentem en la taula que veiem a continuació:

Id.	Regla
1	Horari = 'Matí' => Classe 0
2	Horari = 'Migdia' => Classe 1
3	Horari = 'Tarda' & Nivell físic = 'Alt' => Classe 0
4	Horari = 'Tarda' & Nivell físic = 'Normal' => Classe 1
5	Horari = 'Tarda' & Nivell físic = 'Baix' => Classe 1

Per a trobar la capacitat predictiva de cada branca, hem de contrastar cadascuna de les regles obtingudes que resulten de fer un recorregut per cadascuna de les branques des de l'arrel fins a la fulla corresponent, contra un conjunt de dades de prova. Aquí el tenim:

Cas	Horari	Sexe	Nivell físic	Classe	Classe predita	Regla
101	Matí	Home	Alt	0	0	1
324	Matí	Home	Baix	1	0	1
5344	Migdia	Home	Baix	1	1	2
23	Matí	Dona	Baix	0	0	1
28	Matí	Home	Normal	1	0	1
29	Matí	Dona	Normal	0	0	1
333	Migdia	Dona	Baix	1	1	2
442	Matí	Dona	Normal	1	0	1
32	Migdia	Home	Baix	1	1	2
112	Migdia	Home	Normal	0	1	2
187	Migdia	Home	Normal	0	1	2
54	Tarda	Home	Baix	1	1	5
588	Tarda	Dona	Alt	0	0	4
6536	Migdia	Home	Baix	0	1	2
72	Matí	Home	Normal	0	0	1
811	Tarda	Home	Normal	0	1	4

Podem tenir una aproximació del valor predictiu de cada regla a partir de calcular la proporció de casos en què la regla ha fet una predicció adequada respecte al total de casos que cobria:

Id.	Regla	Casos coberts	Prediccions incorrectes	Error
1	Horari = 'Matí' => Classe 0	7	3	42%
2	Horari = 'Migdia' => Classe 1	6	3	50%
3	Horari = 'Tarda' & Nivell físic = 'Alt' => Classe 0	0	0	–
4	Horari = 'Tarda' & Nivell físic = 'Normal' => Classe 1	2	1	50%
5	Horari = 'Tarda' & Nivell físic = 'Baix' => Classe 1	1	1	0%

Evidentment, aquesta aproximació a la capacitat predictiva s'ha de prendre amb molta precaució. Per exemple, la regla 5 no és que sigui 100% predictiva. No hi ha prou casos en el conjunt de prova per a cobrir realment totes les regles. El que cal posar en evidència, per exemple, és que la regla 2, amb un 50% de capacitat predictiva, no és gaire bona (podríem tenir el mateix resultat llançant una moneda a l'aire!). La regla 1 tampoc no és gaire millor. La pregunta és si eliminant les branques que corresponen a regles amb poca capacitat predictiva es pot millorar la capacitat global de predicció del model.

L'**error global** de tot el model és la suma ponderada dels errors de totes les fulles de l'arbre; és a dir, l'error de cada fulla multiplicat per la probabilitat que un cas vagi a parar a la partició representada per la fulla.

Fem l'aproximació que la probabilitat és el nombre de casos coberts per la fulla (o regla) corresponent respecte al total de casos d'avaluació. Per a un arbre amb *n* fulles:

$$Error_{Global} = \sum_{i=1}^{n} w_i \cdot error_i ,$$

on cada element de l'expressió es defineix a continuació:

- w_i: és el pes o probabilitat de la fulla; és a dir, la probabilitat que un cas sigui classificat en la partició representada per la branca que acaba en la fulla *i*. En altres paraules, la probabilitat que s'utilitzi la regla *i* per a classificar un cas.
- $Error_i$: és l'error corresponent a la branca que acaba en la fulla *i* o, si es prefereix, l'error de classificació propi de la regla *i*.

En el nostre cas tenim que el nombre de fulles és 5 i, utilitzant els valors calculats anteriorment, podem veure que l'error global d'aquest arbre de decisió és el següent:

$$Error_{Global} = \frac{7}{16}0,42 + \frac{6}{16}0,5 + \frac{0}{16}0,0 + \frac{2}{16}0,5 + \frac{1}{16}0,0 = 0,43 .$$

Aquest valor per a l'error global és una xifra prou alta, i per tant, tenim un error prou elevat.

Es pot observar que les diferents regles obtingudes per a cadascun dels possibles recorreguts té un valor predictiu diferent. És més, segons determinades condicions, eliminant les branques que corresponen a regles amb una capacitat predictiva més petita (el que s'anomena *podar l'arbre*) s'obtenen models globals, arbres sencers amb un comportament predictiu conjunt millor.

2.3. Mètodes de poda

Els mètodes de poda* intenten aconseguir arbres que no tinguin més nivells ni particions dels que calen per a assolir un bon nivell de predicció.

*** En anglès, *pruning*.**

Posarem un dels exemples típics, utilitzant un dels conjunts públics de dades del dipòsit de la UCI a Irvine. Es tracta del problema de la decisió de recomanar l'ús de lents de contacte a diversos pacients segons l'edat que tinguin (categoritzada en 'Jove', 'Prepresbícia' i `Presbícia'), la diagnosi actual ('Miop' i 'Hipermetrop') i l'existència d'astigmatisme o no ('Sí' o 'No'). La recomanació final pot ser portar lents de contacte dures, toves, o no portar-ne. Aquest és l'atribut que cal classificar.

Lectura complementària

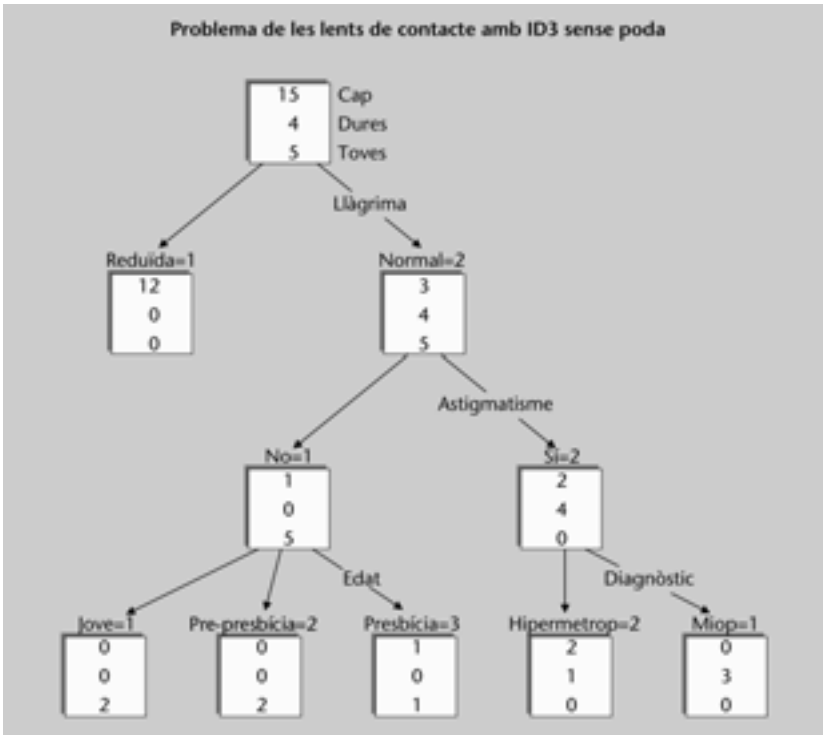
Trobareu el problema que es tracta en aquest subapartat en l'article següent:

J. Cendrowska (1987). "PRISM: an Algorithm for Inducing Modular Rules". *International Journal of Man-Machine Studies* (vol. 4, núm. 27, pàg. 349-370).

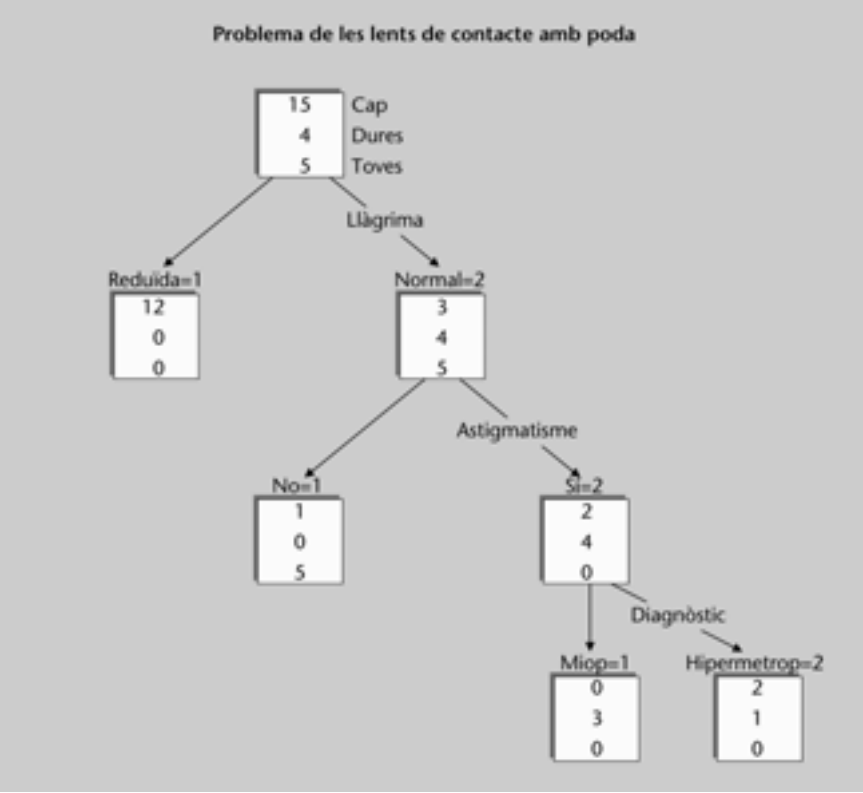
A continuació tenim la taula de vint-i-quatre exemples que cobreix el domini:

Edat	Diagnòstic	Astigmatisme	Llàgrima	Recomanació
Jove	Miop	Sí	Normal	Dures
Jove	Hipermetrop	Sí	Normal	Dures
Prepresbícia	Miop	Sí	Normal	Dures
Presbícia	Miop	Sí	Normal	Dures
Jove	Miop	No	Reduïda	Cap
Jove	Miop	Sí	Reduïda	Cap
Jove	Hipermetrop	No	Reduïda	Cap
Jove	Hipermetrop	Sí	Reduïda	Cap
Prepresbícia	Miop	No	Reduïda	Cap
Prepresbícia	Miop	Sí	Reduïda	Cap
Prepresbícia	Hipermetrop	No	Reduïda	Cap
Prepresbícia	Hipermetrop	Sí	Reduïda	Cap
Prepresbícia	Hipermetrop	Sí	Normal	Cap
Presbícia	Miop	No	Reduïda	Cap
Presbícia	Miop	No	Normal	Cap
Presbícia	Miop	Sí	Reduïda	Cap
Presbícia	Hipermetrop	No	Reduïda	Cap
Presbícia	Hipermetrop	Sí	Reduïda	Cap
Presbícia	Hipermetrop	Sí	Normal	Cap
Jove	Miop	No	Normal	Toves
Jove	Hipermetrop	No	Normal	Toves
Prepresbícia	Miop	No	Normal	Toves
Prepresbícia	Hipermetrop	No	Normal	Toves
Presbícia	Hipermetrop	No	Normal	Toves

En la figura següent veiem un primer resultat d'aplicar el mètode ID3 sense poda:



A cada node apareix la quantitat de casos corresponents a cada classe final, cosa que ens dóna una idea de l'heterogeneïtat de les particions que generen aquest node. Quan apliquem la poda, obtenim un arbre més senzill:



Els mètodes de poda donen arbres més fàcils d'interpretar i més acurats. Potser en aquest exemple la diferència és petita, però en altres que incorporen més atributs acostuma a ser molt més apreciable.

Una manera d'assolir aquest objectiu consisteix a construir primer l'arbre i després analitzar quina és la taxa de predicció que obtenim quan eliminem part de l'arbre. Això s'anomena **mètode de postpoda***.

* En anglès, *post-pruning*.

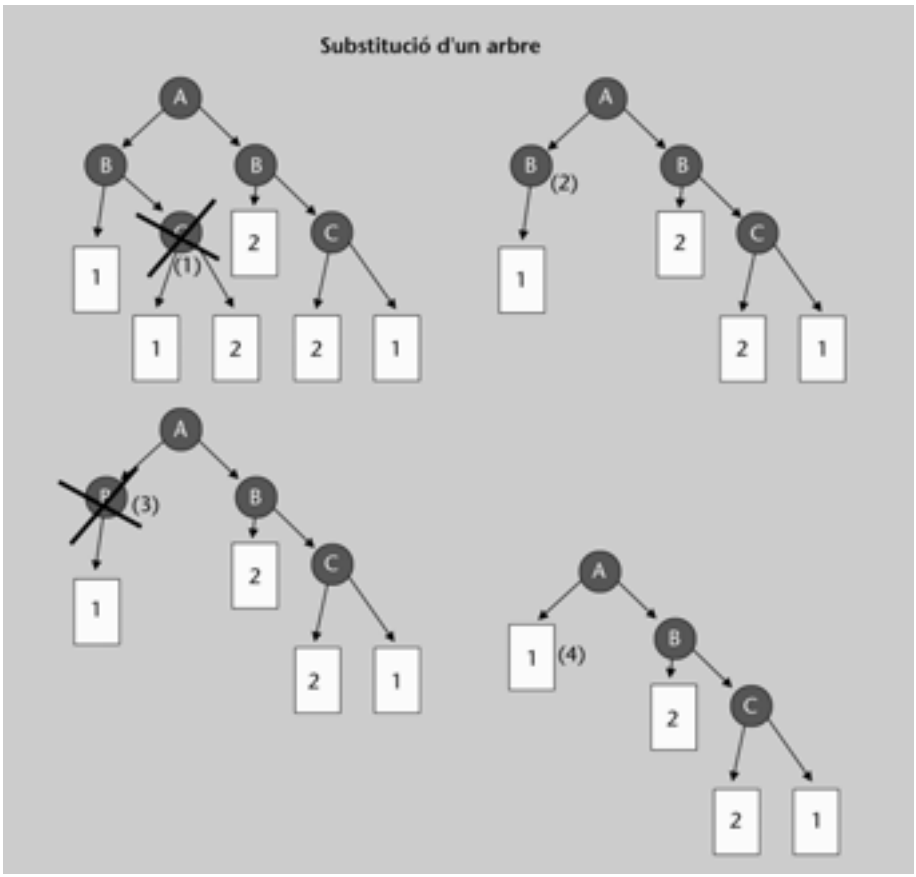
Però pot semblar més útil estalviar-se el procés de construcció dels arbres poc prometedors i preguntar a cada nivell quin és el valor de predicció que ens assegura l'arbre construït parcialment. Si en un determinat node prou informatiu obtenim una avaluació de la taxa de predicció de les particions resultants inferior a un límit de qualitat prefixat, aturem l'expansió de l'arbre. És un **mètode de prepoda***.

* En anglès, *pre-pruning*.

Tots dos mètodes són utilitzats, encara que sembla que la postpoda és més acurada que la prepoda. Les operacions que s'utilitzen en postpoda són la promoció d'un subarbre i la substitució d'un arbre:

a) L'**operació de substitució** consisteix a analitzar el subarbre a partir del node en què el mètode efectua la construcció i substituir-lo per un conjunt de fulles finals. Això comporta establir les categories correctes per a etiquetar les

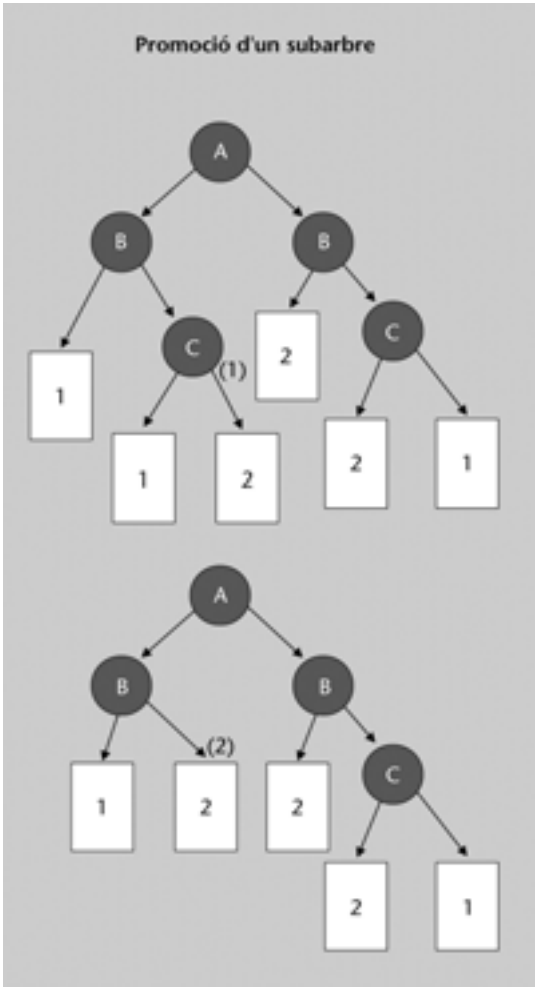
branques. En principi, això podria rebaixar la qualitat final d'un arbre construït amb ID3, que busca tenir fulles tan homogènies com sigui possible, però donaria un arbre més comprensible, més fàcil de calcular i menys sobreespecialitzat. Aquesta darrera característica és important, ja que ens interessa poder generalitzar-lo a nous exemples. La figura següent mostra l'operació de substitució:



A la figura anterior els nombres entre parèntesis indiquen l'ordre en què es consideren els nodes de l'arbre. El subarbre que penja del subarbre esquerre *B* s'ha substituït per una partició en què les observacions s'assignen majoritàriament a la classe 1.

b) L'**operació de promoció** funciona en sentit contrari al de la substitució. Si aquella actuava en sentit descendent, aquesta ho fa de manera ascendent, de les fulles cap a l'arrel. El mètode C4.5 i el seu successor, el C5.0, tots dos molt semblants a ID3, incorporen un mètode de postpoda per promoció.

Amb l'operació de promoció es detecta que un subarbre no és útil o interessant (per exemple, té una capacitat de predicció baixa) i se substitueix per un subarbre que ofereixi expectatives millors. És clar, aquesta operació implica una reclassificació d'exemples. Tots els exemples que eren "sota el paraigua" del node original, cal ubicar-los sota el nou i, per tant, cal crear particions noves. La complexitat d'aquesta operació no és negligible.



Tot el subarbre que depenia del node esquerre amb l'etiqueta *B* se substitueix pel subarbre que penja de *C*, i cal reclassificar les observacions de les particions de *B* i de *C*.

La manera més senzilla de càlcul per a decidir que cal efectuar la promoció o substitució d'un arbre consisteix a anticipar la taxa d'error que s'obté amb substitució (o promoció) i sense.

Ja hem dit que, cada cop que ens plantegem canvis d'aquesta mena, cal reclassificar les observacions que queden per sota del node subjecte d'anàlisi.

Comentarem dos mètodes de poda: el típic C4.5 i un mètode basat en el principi MDL.

2.3.1. Poda amb el mètode C4.5

Per a calcular la taxa d'error sorgeix una situació de compromís. El més correcte seria emprar un conjunt de dades diferent del que emprem per a construir l'arbre; és a dir, un conjunt diferent del d'entrenament. Altrament, com a mínim es poden reservar algunes de les dades del conjunt original amb aquest fi.

Finalment, i és el que fa el mètode C4.5, es poden utilitzar les mateixes dades del conjunt d'entrenament. Contra tota lògica, funciona prou bé.

Abans, però, haurem de fer una excursió estadística per a conèixer una altra distribució de probabilitat.

La distribució binomial

El problema del càlcul de la taxa d'error (o, a l'inrevés, de la precisió) d'un mètode de classificació es pot entendre com el de l'estimació d'una proporció en una població.

Volem saber si, vist que hem calculat a partir de les dades una taxa d'error del e%, quina és la taxa que podríem anticipar en exemples nous procedents de la mateixa població.

En principi hauríem de tenir preferències respecte a les taxes calculades a partir de mostres més grans. Això és particularment rellevant quan cal comparar les taxes d'error en diversos nivells d'un arbre, ja que, com més avall de l'arbre ens trobem, menys observacions hi ha (més petites són les particions).

La predicció de la taxa d'error vista d'aquesta manera es pot entendre com el càlcul de la proporció d'errors o èxits que farà un determinat procés de classificació respecte a una població.

El resultat del model de classificació per a cada observació pot ser correcte o incorrecte (ben classificat, mal classificat). En relació amb l'observació, doncs, el procés té dos resultats: cert o fals, correcte o incorrecte, cara o creu. Sobre un conjunt de n observacions, efectuarem una seqüència de n proves que poden donar com a resultat correcte o incorrecte. Volem trobar quina és aquesta proporció.

Aquesta mena de processos, que són anàlegs a llançar n vegades una moneda a l'aire i intentar inferir quina és la proporció de cares i creus que esperem veure (el 50%, si la moneda no està trucada), segueix una **distribució binomial** o **distribució de Bernoulli**.

El problema d'estimació que ens plantegem davant de fenòmens com el de llançar una moneda a l'aire o classificar observacions és estimar, a partir de la proporció observada p, quina seria la proporció real. Per tant, ens trobem davant un problema típic d'estimació. Un altre cop, haurem d'establir intervals de confiança i decidir on se situa la proporció en la població. Però ara la distribució no és normal, sinó binomial.

Aquesta distribució, per a una taxa d'èxit de p, segueix una llei de probabilitat que té de mitjana p i de variància $1 - p$. En un procés de n proves d'aquesta mena, on hem observat k èxits (classificacions correctes), la variable aleatòria p', que representa la proporció d'èxits esperats:

$$p' = \frac{k}{n}$$

segueix una llei binomial amb mitjana p i variància $\frac{p(1-p)}{n}$.

Quan n és gran, això s'aproxima a la distribució normal. Per tant, per a trobar un interval de confiança del 90%, ens interessa trobar el valor z tal que:

$$P[-z \leq p' \leq z] = 90\%.$$

Es pot comprovar que el valor corresponent de z és 1,65.

Per al cas que ens interessa, hem de normalitzar la variable p' perquè tingui mitjana 0 i desviació típica 1. Hi apliquem la transformació que ja coneixem i en resulta:

$$P\left[-z \leq \frac{p' - p}{\sqrt{p(1-p)/n}} \leq z\right].$$

Per a trobar els límits de l'interval de confiança, fem el de sempre: consultem les taules corresponents i després reconvertim el valor obtingut al rang de valors de p' mitjançant la transformació següent (que no ens entretindrem a derivar):

$$p = \left(k + \frac{z^2}{2n} \pm z\sqrt{\frac{k}{n} - \frac{k^2}{n} + \frac{z^2}{4n^2}}\right) \Big/ \left(1 + \frac{z^2}{n}\right).$$

Encara hem de veure com es relaciona això amb C4.5. En aquest cas, intentem estimar el contrari de p, que era la proporció esperada d'èxits. Recordem que volem estimar la proporció d'error. La cosa és fàcil, consisteix a fer el següent:

$$\% \text{ d'èxits} + \% \text{ d'errors} = 1.$$

En conseqüència, es tracta d'estimar la proporció de $e = (1 - p)$. Una última modificació molt important és que ara treballem directament amb les dades d'entrenament; per tant, fem una estimació pessimista de e, agafant el nivell superior de confiança. El nivell de confiança que pren l'algorisme C4.5 per defecte és el 25%. Per tant, es tracta de trobar (normalitzant un altre cop per a obtenir una distribució amb mitjana 0 i desviació estàndard 1):

$$P\left[\frac{k - e}{\sqrt{e(1-e)/n}} > z\right]$$

on k ara és la taxa d'error observada.

Lectura complementària

Amb relació a fer una estimació pessimista de e, trobareu més informació en l'article següent:

J.R. Quinlan (1987). "Simplifying Decisions Trees". *International Journal of Man-Machine Studies* (núm. 27, pàg. 221-234).

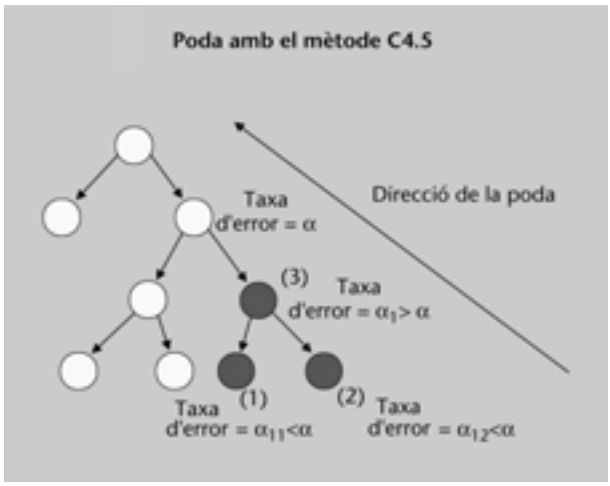
La *z* corresponent al nivell de confiança del 25% és 0,69. Per a veure què val això en termes de proporcions, hem de desfer el canvi de variable i, tenint en compte que ara calculem la proporció d'error, en resulta una expressió un xic diferent de la d'abans:

$$p = \frac{\left(k + \frac{z^2}{2n} \pm z\sqrt{\frac{k}{n} - \frac{k^2}{n} + \frac{z^2}{4n^2}}\right)}{1 + \frac{z^2}{n^2}}$$

Després d'això, si apreciem, per exemple, una proporció d'error del 15% a partir de cent mil observacions, podem dir que la proporció veritable d'error oscil·larà entre el 12,09 % i el 16,03%. Es tracta de fer el següent:

1) Per a cada node, observem la proporció d'error (*e*) i calculem la proporció segons l'estimació pessimista (*e'*); després combinem aquests estimadors ponderant-los segons el nombre de valors de cada partició.

2) Si l'estimació pessimista per a un node és més petita que la dels fills, llavors eliminem els fills.



2.3.2. Poda amb el mètode MDL

S'ha comprovat experimentalment que els mètodes de poda basats en l'estimació pessimista que hem descrit generen arbres massa grans i taxes d'error més altes del que és necessari. Una manera alternativa de plantejar-se la poda d'arbres i que genera arbres més compactes és la que utilitza el mètode de mínima descripció de longitud. Unes altres proven de crear diversos arbres respecte a diferents conjunts de dades i després avaluen amb validació creuada (que es veu en un altre mòdul) els diversos models i es queden amb el millor (amb l'increment consegüent de la complexitat de tot el procés).

Lectures complementàries

Vegeu els resultats experimentals dels mètodes de poda basats en l'estimació pessimista i el mètode de mínima descripció de longitud, respectivament, en les obres següents:

J. Mingers (1989). "An Empirical Comparison of Pruning Methods for Decision Tree Induction". *Machine Learning* (núm. 4, pàg. 227-243).

M. Rissanen (1985). "The Minimum Description Length Principle". A: S. Kotz; N.L. Johnson (ed.). *Encyclopedia of Statistical Sciences* (vol. 5). Nova York: John Wiley & Sons.

Tot seguit donem l'explicació del mètode de mínima descripció de longitud:

1) La **formalització del problema** en el marc de l'MDL és la que presentem a continuació:

Suposem que tenim un atribut de classe, c_t, que pot prendre 0, ..., $m - 1$ valors. El conjunt de dades està format per una successió de n observacions. Cada una té els valors corresponents als diversos atributs X_1, ..., X_n més el que correspon a l'atribut de classe. Suposarem que cada atribut pren valors en un conjunt de valors determinat. La variable categòrica X_i pren valors x_i en un conjunt de valors $\{1, \ldots, r_i\}$.

El **mètode de mínima descripció de longitud** (MDL) intenta trobar el model dins de cada classe que permeti la mínima codificació de la seqüència de la classe c_n donats els valors observats.

Cal trobar, dins l'espai de possibles subarbres que es poden generar a partir d'un arbre T, els mínims. A cada node de l'arbre T hi ha un atribut amb el nombre d'observacions corresponents i el valor de tall per a l'atribut corresponent dins l'arbre. D'aquesta manera, haurem de codificar per a cada node els valors dels atributs per a cadascun dels subarbres que genera (o el punt de tall corresponent en variables contínues) i les probabilitats de la classe.

2) La codificació del problema es basa a fer servir com a codi la longitud de la cadena necessària per a representar cada subarbre amb la convenció de guardar, per a cada node, la informació que hem comentat. La fórmula per a calcular aquesta longitud té un aspecte força impressionant:

$$\log_2 n_i + \log_2 \frac{t!}{t_0! + \ldots + t_n!} + \log_2 \binom{t + m - 1}{m - 1}$$

on n_i és el nombre de vegades que el símbol i (un valor d'un atribut categòric, per exemple) apareix en la seqüència de variables de classe c_t. Recordem que hi pot haver t classes. El tercer terme representa la longitud de codificació necessària per a codificar el nombre de vegades que apareix cada valor i es pot considerar el cost del model per al model (subarbre) de la classe corresponent. Els termes primer i segon representen la longitud de codificació necessària per a codificar les dades observades. L'inconvenient principal és que tots dos termes s'aproximen al mateix ordre de magnitud quan alguns dels comptatges d'aparicions de valors són propers a 0 o al nombre de classes, t.

Una millora d'aquest model és considerar la longitud següent:

$$L(c_t) = \sum t_i \log_2 \frac{t}{t_i} + \frac{m-1}{2} \log_2 \frac{t}{2} + \log_2 \frac{\pi^{m/2}}{\Gamma(m/2)} .$$

on Γ és la funció gamma i t_i representa el nombre de vegades que un valor apareix dins la seqüència de classe c_t. Aquesta longitud de codificació, anomenada *complexitat estocàstica*, té propietats d'optimació ben provades.

La decisió de podar el subarbre que penja d'un node determinat es pren en aquest mètode utilitzant com a criteri la longitud de descripció dels diversos subarbres considerats. Si per a un node determinat, a, la longitud de descripció és l_a i la longitud ponderada dels seus k subarbres a_1, ..., a_k és:

$$l_{fills} = \frac{1}{n}\sum_{i=1}^{k} l_i ,$$

llavors es podaran els subarbres si l_{fills} és més petita que l_a.

Lectures complementàries

Per a una discussió més detallada i la comparació de resultats amb relació al mètode de mínima descripció de longitud, consulteu les obres següents:

R.E. Krichevsky; V.K. Trofimov (1983). "The Performance of Universal Coding". *IEEE Transactions on Information Theory* (vol. IT-27, núm. 2).

M. Rissanen (1989). *Stochastic Complexity in Statistical Inquiry*. Nova Jersey: World Scientific Publishers Company.

3. Mètodes de construcció d'arbres de decisió per a regressió i classificació (CART)

El mètode CART és un algorisme de construcció d'arbres de decisió proposat el 1984 per Breiman i els seus col·laboradors. *CART* és la sigla de *classification and regression trees* (que vol dir 'arbres de classificació i regressió'). La majoria de paquets comercials de construcció d'arbres de decisió o de construcció de regles de decisió utilitzen d'una manera més o menys explícita aquest algorisme.

Lectures complementàries

Trobareu la presentació de l'algorisme CART en l'obra següent:

L. Breiman; H. Friedman; R. Olshen; C. Stone (1984). *Classification and Regression Trees*. Belmont: Wadsworth.

Les diferències principals d'aquest mètode respecte al mètode ID3 es concentren en els aspectes següents:

a) El tipus d'arbre que es construeix: en l'algorisme CART els arbres que es construeixen són binaris; a cada node hi ha un punt de tall (per un procediment semblant al que s'ha explicat per a trobar punts de tall en la discretització d'atributs continus) que separa en dos el conjunt d'observacions.

b) Els tipus d'atributs: en principi, l'algorisme CART pot treballar amb atributs continus (tot i que les modificacions de ID3 també ho poden fer).

c) Pot fer tant classificació com regressió: en el primer cas, la variable per a predir ha de ser categòrica amb un valor per a cada classe possible.

d) Mesura d'homogeneïtat i criteri d'aturada en el procés de partició i divisió de l'arbre: en aquest mètode és la **mesura de Gini**, encara que hi ha variants que n'escullen d'altres.

El procediment de construcció de l'arbre en el cas del CART passa per considerar en cada moment el fet de trobar l'atribut que actua com a millor separador* o punt de tall.

*** En anglès, *splitter*.**

Per a fer-ho utilitza la **mesura de Gini** com a índex de diversitat de cada partició possible. La mesura de Gini estableix que el millor separador és l'atribut que redueix la diversitat de les diferents particions. Per tant, el que fa CART és maximitzar-ne la diferència:

$$\text{Mesura de Gini} = \text{diversitat abans de la partició} - \text{diversitat en el subarbre esquerre} + \text{diversitat en el subarbre dret}.$$

S'escull, doncs, el millor separador i es converteix en l'arrel de l'arbre. Això genera dues particions i es torna a procedir amb cadascuna de la mateixa manera.

Si en aquest punt algun atribut pren només un sol valor respecte a tots els elements de la partició, el deixem de considerar. Quan no es poden trobar més separadors, el node s'etiqueta com una fulla terminal. Quan totes les particions corresponen a fulles terminals, s'ha acabat de construir l'arbre.

La poda a CART s'efectua en tres passos:

1) Generar diversos subarbres podats "interessants".

2) Obtenir estimacions de l'error de cadascun d'aquests subarbres.

3) Escollir el subarbre que presenti la millor estimació.

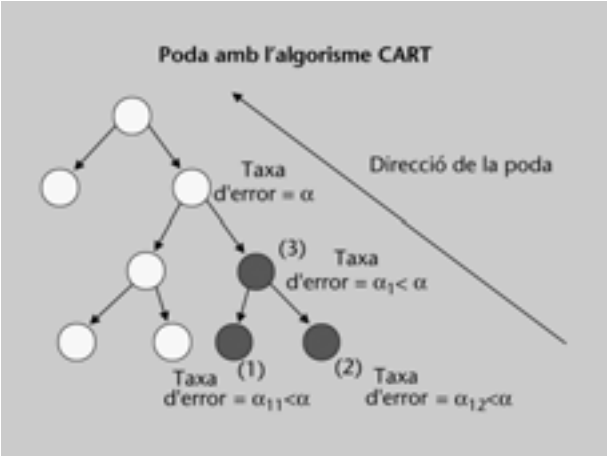
El primer pas consisteix a generar seqüències de subarbres niades:

$$arbre_1, \ldots, arbre_n.$$

A cada pas *i* el node que cal podar s'obté de l'arbre predecessor en la seqüència $arbre_{i-1}$. Per a escollir-lo, es fa servir la taxa d'error ajustada, *e'*, del subarbre corresponent:

$$e' = e_{arbre_i} + \alpha \cdot nfulles(arbre_i)$$

on e_{arbre_i} és la proporció d'error observat en l'arbre *i*, $nfulles(arbre_i)$ és el nombre de fulles de $arbre_i$ i α és un coeficient que té un comportament interessant. Per a trobar el primer subarbre, es calculen les taxes d'error ajustades de tots els subarbres que contenen l'arrel incrementant gradualment α. Quan la taxa d'error ajustada d'alguns dels subarbres és més petita o igual que la de tot l'arbre, ja tenim un primer subarbre candidat $arbre_1$ per a ser podat. Llavors es poden totes les branques que no són part de $arbre_1$ i el procés es repeteix. D'aquesta manera, començant per les fulles, es va procedint cap a l'arrel de l'arbre. Gràficament, el procés és el següent:



Els nombres indiquen en quin ordre es consideren els nodes. Els que surten de color negre són candidats a ser podats. Noteu que els dos nodes del subarbre de la dreta no són candidats a ser podats fins que el seu pare mostra una taxa inferior a α.

Ara aplicarem CART a un altre conjunt de dades prou conegut, també procedent del dipòsit de la UCI a Irvine. És el conjunt IRIS, que guarda dades sobre la classificació de les diverses espècies d'iris (lliris), existents segons els paràmetres següents: l'amplada del pètal (*Petal width*), la longitud del pètal (*Petal length*), l'amplada del sèpal (*Sepal width*) i la longitud del sèpal (*Sepal length*).

La variable de classificació és la classe de la planta, evidentment. N'hi ha tres: *Iris versicolor*, *Iris setosa* i *Iris virginica*.

Classe	Longitud del sèpal	Amplada del sèpal	Longitud del pètal	Amplada del pètal
Iris setosa	5,1	3,5	1,4	0,2
Iris setosa	4,9	3	1,4	0,2
Iris setosa	4,7	3,2	1,3	0,2
Iris setosa	4,6	3,1	1,5	0,2
Iris versicolor	7	3,2	4,7	1,4
Iris versicolor	6,4	3,2	4,5	1,5
Iris virginica	5,8	2,7	5,1	1,9
Iris virginica	7,1	3	5,9	2,1
Iris virginica	6,3	2,9	5,6	1,8
Iris versicolor	6,9	3,1	4,9	1,5

En aquest problema totes les dades són numèriques i el que troba CART a cada pas és el següent:

- l'atribut més discriminant;
- el punt de tall que garanteixi particions uniformes per a l'atribut.

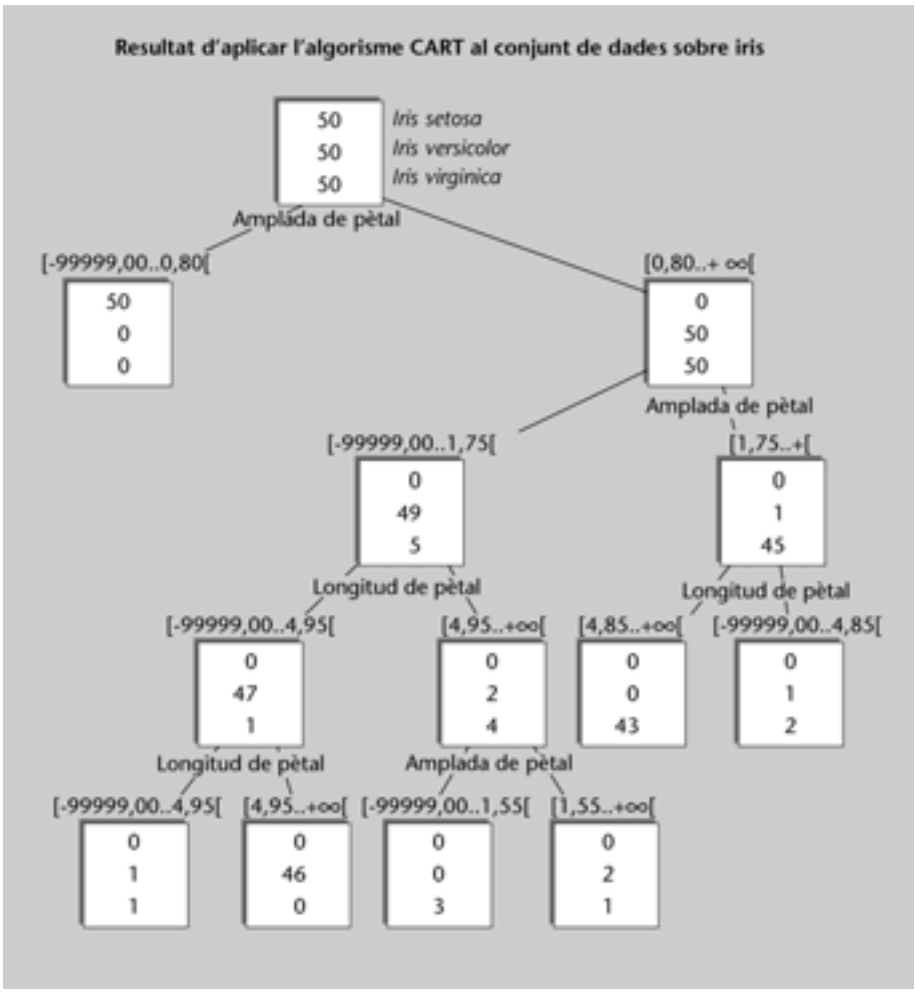
Amb això, a cada node s'efectua un test que compara els valors de la variable de què es tracti amb el punt de tall, de manera que se separen les dades en dues particions: les que tenen un valor inferior al punt de tall i les que en tenen un de superior.

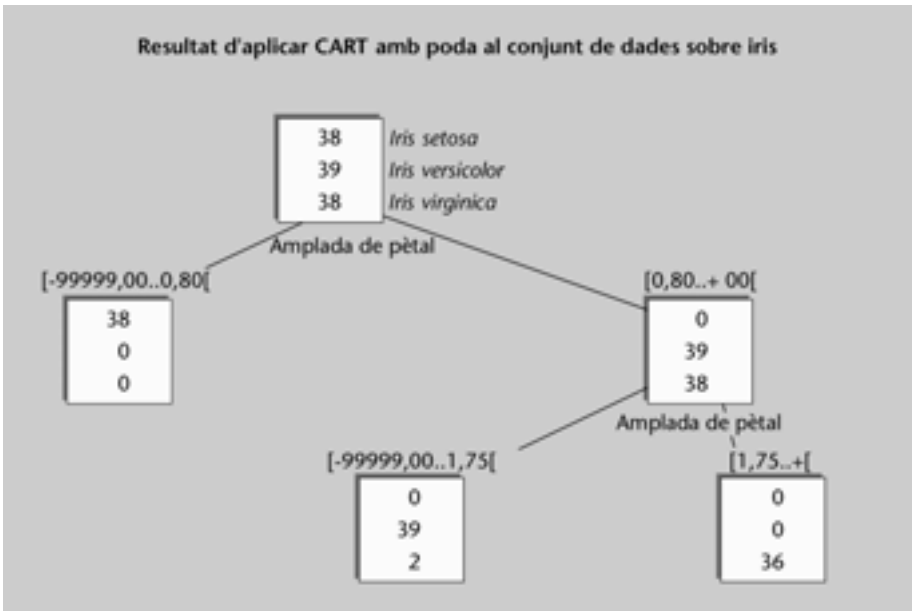
En la figura de la pàgina següent podeu veure el resultat d'aplicar l'algorisme CART al conjunt de dades IRIS. Observeu en aquest exemple que cada subarbre és binari:

Lectures complementàries

Per a una comparació de mètodes de poda, consulteu l'article següent:

F. Esposito; D. Malerba; G. Semeraro (1997). "A Comparative Analysis of Methods for Pruning Decision Trees". *IEEE Transactions on Pattern Analysis and Machine Intelligence* (vol. 19, pàg. 476-493).


Resultat d'aplicar l'algorisme CART al conjunt de dades sobre iris
50
50
50
Iris setosa
Iris versicolor
Iris virginica
Amplada de pètal
[-99999,00..0,80[
50
0
0
[0,80..+ ∞[
0
50
50
Amplada de pètal
[-99999,00..1,75[
0
49
5
[1,75..+[
0
1
45
Longitud de pètal
Longitud de pètal
[-99999,00..4,95[
0
47
1
[4,95..+∞[
0
2
4
[4,85..+∞[
0
0
43
[-99999,00..4,85[
0
1
2
Longitud de pètal
Amplada de pètal
[-99999,00..4,95[
0
1
1
[4,95..+∞[
0
46
0
[-99999,00..1,55[
0
0
3
[1,55..+∞[
0
2
1


Resultat d'aplicar CART amb poda al conjunt de dades sobre iris
38
39
38
Iris setosa
Iris versicolor
Iris virginica
Amplada de pètal
[-99999,00..0,80[
38
0
0
[0,80..+ 00[
0
39
38
Amplada de pètal
[-99999,00..1,75[
0
39
2
[1,75..+[
0
0
36

4. Mètodes de construcció d'arbres de decisió per a predicció numèrica (CHAID)

El mètode CHAID de construcció d'arbres de decisió és un algorisme proposat per Hartigan el 1975. Procedeix d'un mètode anterior, AID, que intentava trobar relacions estadístiques significatives entre variables (*AID* és la sigla de l'expressió anglesa *automatic interaction detection*, i significa 'detecció automàtica d'interaccions'). La manera de detectar les interrelacions entre les variables passa per construir un arbre de decisió.

Lectures complementàries

Trobareu la presentació del mètode CHAID en l'obra següent:

J.A. Hartigan (1975). *Clustering Algorithms*. Nova York: John Wiley & Sons.

Les diferències principals del mètode CHAID respecte als altres mètodes rauen en els aspectes següents:

- El mètode CHAID està restringit a variables categòriques.

- El mètode atura la construcció de l'arbre en el moment en què detecta que es pot produir una sobreespecialització.

El criteri d'establiment del punt de tall en cada node és el mateix que s'utilitza en el mètode *chi merge* per a efectuar la discretització de variables contínues. La poda es duu a terme efectuant una prova de significació als errors esperats de cada subarbre.

5. Mètodes de construcció d'arbres de decisió multivariants (LMDT)

Hem comentat abans que un dels problemes que generen certs mètodes de construcció d'arbres és la complexitat de l'estructura resultant, que provoca una transformació que porta a regles poc entenedores. Ja hem vist que la poda pot resoldre aquesta mena de problemes, introduint certs augments de cost computacional. Una altra manera de reduir el problema és preguntar-se a cada node no solament pel valor d'un únic atribut, sinó també pels valors de més d'un atribut al mateix temps. Això, a més, té un gran avantatge afegit, i és que es poden resoldre problemes de classificació als quals els arbres basats en particions binàries o en un únic valor no poden donar solució.

L'objectiu dels mètodes LMDT...

... és construir arbres més senzills, igualment predictius i que puguin resoldre problemes més complexos.

El problema de la complexitat excessiva del model sorgeix quan es disposa de conjunts de dades on les diverses classes no generen regions amb límits que es puguin definir amb línies rectes o zones de l'espai que no es poden descriure amb equacions lineals. Aquest problema es coneix com a **problema de les regions de classificació no linealment separables** i afecta tots els mètodes de classificació.

Vegeu el mòdul "Cassificació; xarxes neuronals" d'aquesta assignatura.

La diferència principal d'aquest mètode respecte als altres rau en el fet que, en comptes de considerar un únic atribut cada cop per a efectuar la separació, se'n consideren *n*. Donat un node de decisió, en comptes d'aplicar la prova tradicional al grau de separabilitat que indueix un únic atribut, l'algorisme aplica el test a més d'un atribut al mateix temps. Això es fa per a resoldre un problema típic de la classificació com la dificultat o impossibilitat de representar particions que no són ortogonals als eixos de l'espai de què es tracti.

Hem d'entendre que el conjunt d'atributs que representa el domini defineix un espai de tantes dimensions com atributs hi ha. Per a cada atribut, X_1, ..., X_n, definim un eix. Una observació es converteix en un punt en aquest espai.

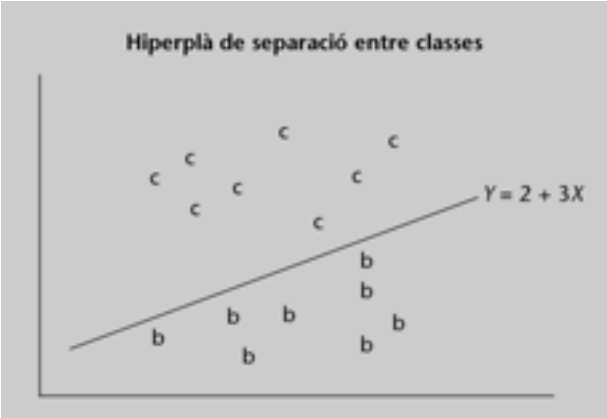
Exemple d'espai de dues dimensions

Suposem que només representem la renda i l'edat dels socis d'Hyper-Gym. Això defineix un espai de dues dimensions. L'observació (5.000.000,40) corresponent a una renda de 5.000.000 i una edat de quaranta anys és un punt representat en un espai o, per exemple, l'eix *X* correspon a l'atribut *Renda*, i l'eix *Y*, a l'atribut *Edat*. Podeu estendre aquest raonament a més atributs i, per tant, a més dimensions. Aquesta és una idea força potent que retrobarem contínuament. Hi dedicarem més espai en parlar de mètodes d'agregació.

El **problema de la separabilitat** consisteix en el fet que, per a certs conjunts d'observacions, és impossible definir línies (o hiperplans) en l'espai corresponent que destriïn perfectament les observacions que corresponen a una classe i a una altra.

5.1. Tractament dels conjunts linealment separables

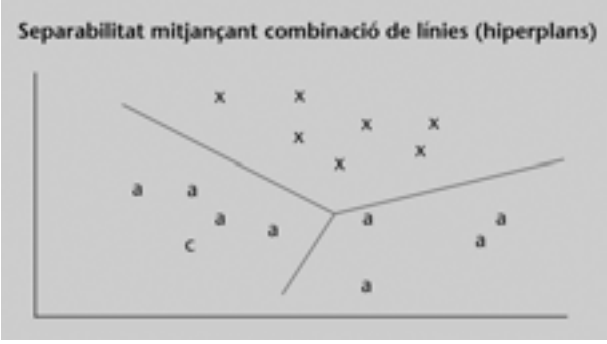
A la figura següent tenim observacions de dues classes que es poden separar descrivint la línea que els separa mitjançant una equació linial.



Si utilitzem només un test respecte als valors d'un únic atribut, no es poden separar les classes correctament. El que sí que es pot fer mitjançant els tests tradicionals és trobar condicions per a separar les particions quan aquestes són ortogonals als eixos (en dues dimensions, quan els dominis de les classes són quadrats).

En el cas d'aquestes particions, les línies que les descriuen (sobre el pla *X*-*Y*) tenen una expressió del tipus $y \leq N$. Quan establim una comparació en un node de decisió, imposem o descobrim quin valor de la variable que ens interessa (l'atribut del node de decisió) permet d'englobar tots els casos que pertanyen a la mateixa classe sota la recta que es determini. En la figura anterior es pot veure que la partició que identifica els objectes etiquetats amb una *x* correspon a la regió $Y \geq 2$ i $X \leq 2$, que és una regió quadrangular, no és possible trobar-hi línies tan senzilles com aquestes per a aïllar la regió de l'espai bidimensional.

Quan, en comptes d'un espai de dimensió 2 ens trobem amb espais de dimensionalitat més alta, hem de definir hiperplans ortogonals en els diversos eixos. Evidentment, si el conjunt de casos no ocupa una regió que es pugui caracteritzar mitjançant hiperplans ortogonals, llavors cal aproximar la regió per mitjà d'una altra mena de plans.



En aquest cas, la línia que es pot dividir les dues regions, les dels objectes etiquetats com a *a* i la dels etiquetats com a *b,* es pot separar mitjançant una línia diagonal que és una "combinació linial" de *Y i de X*.

En el cas d'espais de dimensió més gran , cal fer una aproximació a la forma en l´espai de cada partició, trobant les combinacions lineals de diversos atributs.

$$Y = a_1x_1 + a_2x_2 + \ldots + a_nx_n.$$

El problema, és clar, és encertar quins són els valors a_i, els pesos, per a poder isolar adequadament cadascuna de les particions. El que està clar, però, és que cal usar simultàniament els valors de més d'un atribut.

Una proposta de millora per a aquest problema és la que aporta el sistema LMDT, proposada per Utgoff el 1986.

LMDT és la sigla de l´expressió anglesa Linear Machine Decision Trees

El **mètode LMDT de construcció d'arbres de decisió multivariants** assigna una classe a cada node com a resultat del test respecte a diversos atributs. En comptes de preguntar per un únic valor en cada node, es fa un test respecte al conjunt d'atributs.

L'algorisme que efectua aquest test construeix un conjunt de funcions linealment discriminants, de manera que cadascuna d'aquestes funcions es fa servir després per a assignar cada observació a la classe que li correspon. Per a tot node que no és una fulla, es defineixen tantes funcions discriminants com classes hi hagi. Si hi ha *n* classes, s'han de definir *n* funcions discriminants lineals per a tot node del subarbre.

Suposada una observació *Y* definida en termes dels atributs que defineixen el domini $X_1, \ldots, X_n$: $Y = x_1, \ldots, x_n$, on les lletres en minúscula representen els valors que prenen les variables corresponents, el mètode LMDT actua sobre valors continus i normalitzats.

Una funció discriminant es defineix de la manera següent: $g_{1(Y)}W_i^TY$ on *W* és un vector de coeficients ajustables que anomenarem **pesos**.

Donat un node (que no sigui una fulla) que contingui un conjunt de funcions discriminants, $g_1, \ldots, g_n$, i donat un conjunt de possibles classes, $c_1, \ldots, c_n$, llavors:

$$Y \in c_i \Leftrightarrow g_i(Y) > g_j(Y), \forall i, 1 \leq i \leq n, i \neq j.$$

És a dir, una observació pertany a una classe c_i quan el producte interior amb el vector de pesos W_i és màxim respecte a la resta de funcions discriminants (recordem que n'hi ha una per a cada classe).

Per tant, és important poder extreure a partir del conjunt de dades originals els pesos W_i que assegurin que obtenim el millor arbre possible.

L'aprenentatge pròpiament dit consisteix a ajustar els vectors W corresponents a les funcions discriminants, g_i, augmentant els pesos de W_i, on i és la classe a la qual pertany una observació, i disminuint els pesos de W_j, on j seria una classe errònia per a l'observació que tractem en un moment determinat.

Donat un pes W_i, en veure una nova observació Y, hem d'efectuar un ajust que formalitzem així:

$$W_i \leftarrow W_i + cY$$
$$W_j \leftarrow W_j + -cY, (\forall i, 1 \leq i \leq n, i \neq j)$$

on j és l'índex que representa les classes errònies per a l'observació que considerem i c és un factor de correcció que es calcula de manera que tota observació que s'hagi classificat erròniament ara es classifiqui correctament.

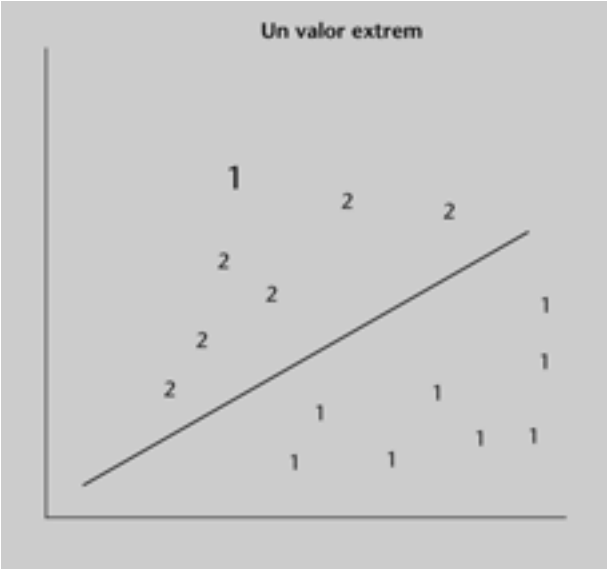
5.2. Tractament dels conjunts no linealment separables

Per al cas dels **conjunts de dades no linealment separables**, el sistema LMDT introdueix correccions contínues en els errors de classificació i el procés no s'estabilitza o fa que la classificació sigui imprevisible. Cal assegurar un comportament estable en aquestes situacions. La solució consisteix a introduir una modificació en el paràmetre de correcció c.

La idea és parar més atenció als errors de classificació més grans (es veu una idea semblant en el cas de la retropropagació en xarxes neuronals).

Vegeu la retropropagació en l'apartat 3 del mòdul "Classificació: xarxes neuronals" d'aquesta assignatura.

La figura de la pàgina següent intenta explicar la situació que acabem de presentar:



L'observació destacada en la cantonada superior esquerra requeriria un ajust força important en la línia que estableix la decisió per a poder-la classificar correctament. Però això seria negatiu respecte a la situació actual, tenint en compte que la línia definida actualment ja es pot considerar una línia de separació prou bona.

La traducció d'aquesta intuïció a termes formals introdueix el factor de correcció següent:

$$c = \frac{1}{1+k}$$

on k és el factor de la correcció necessària per a ajustar els vectors de pesos de manera que una observació classificada incorrectament passi a estar-ho correctament. Per a assolir l'estabilitat en el comportament del classificador, cal anar reduint gradualment l'atenció pel que fa als errors importants amb el factor:

$$c = \frac{\beta}{\beta+k}$$

i anar atenuant el factor β durant l'entrenament. Aquest valor està fixat per defecte en 0,995, però l'usuari del sistema el pot modificar. Això és equivalent a un procés de *simulated annealing*, que cerca l'estabilitat del procés fent-lo convergir cap a una solució estable.

Un segon problema és el de les observacions que són massa a prop de la frontera entre classes. Aquestes observacions fan que el problema es converteixi en no separable. En la figura de la pàgina següent tenim la situació expressada gràficament:

En aquest cas, k s'aproxima a 0, i això fa que c s'aproximi a 1, independentment del valor que prengui β. En aquest cas cal atenuar el factor de correcció c independentment de k. Això s'aconsegueix multiplicant el coeficient anterior per β. Aleshores, el nou coeficient c queda de la manera següent:

$$c = \frac{\beta^2}{\beta + k}$$

L'algorisme que segueix el mètode LMDT introdueix l'atenuació de β només quan la magnitud dels vectors w_i s'ha reduït en el pas actual d'ajust de pesos, però s'havia incrementat en l'ajust de pesos que es va fer en la iteració anterior.

La **magnitud dels vectors** es defineix com la suma de les magnituds de cadascun d'aquests.

Aquest criteri s'ha adoptat davant l'observació que la magnitud dels vectors s'incrementa ràpidament durant el començament de l'entrenament i s'estabilitza quan la línia de decisió s'apropa a la seva ubicació final.

El procés LMDT, doncs, comprèn els passos següents:

1) Assignació de valors inicial als vectors de pesos de les 1, ..., n classes per al node inicial.

2) Ajust de valors dels pesos segons les classificacions correctes.

3) Detecció de les variables que contribueixen menys a la discriminació dels conjunts d'observacions corresponents a cada branca.

4) Eliminació de les variables menys discriminants.

El procés es repeteix per als nodes que s'obren a l'extrem de cada branca.

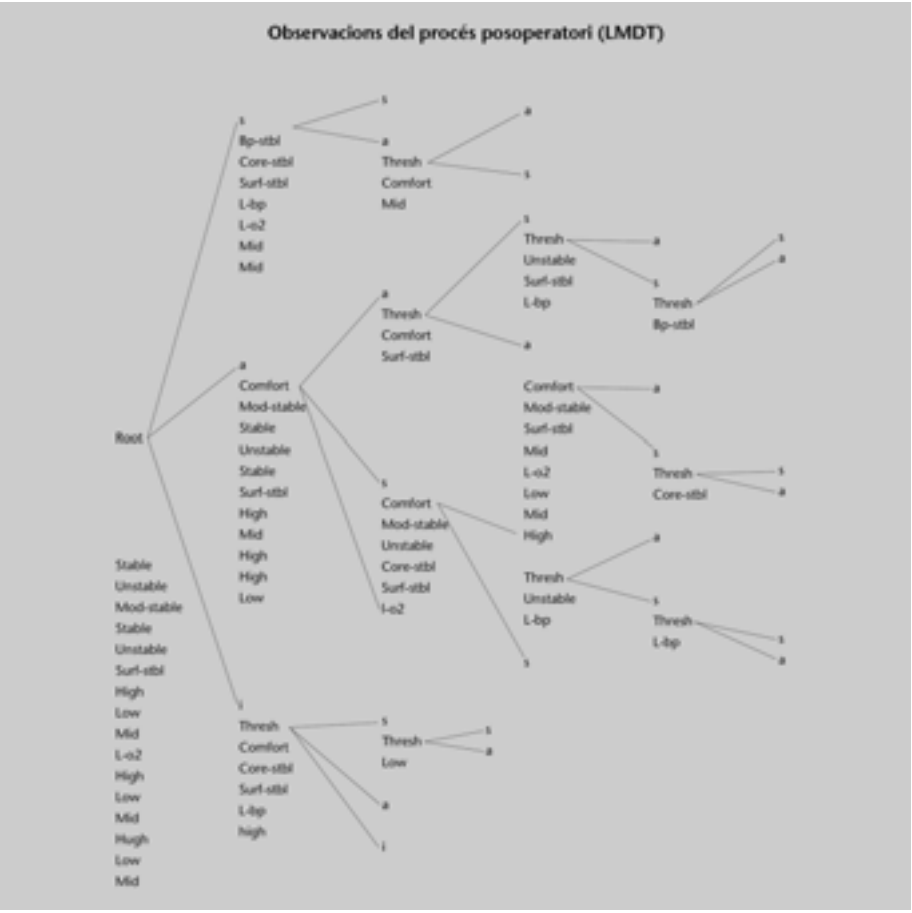
L'**eliminació** consisteix a provar d'eliminar els atributs un per un fins que un redueix la precisió de classificació del node (mesurada com a proporció d'observacions classificades correctament).

La **mesura de la contribució d'un atribut** es calcula en funció de la dispersió que genera en els pesos de cadascuna de les classes. Un pes important contribueix positivament a la funció de discriminació per a l'atribut corresponent i, per tant, a la seva capacitat de discriminació. Un atribut que té els seus pesos uniformement distribuïts entre les classes possibles és poc discriminant i, per tant, contribueix poc a les funcions discrimants de cada classe.

El **criteri d'eliminació d'atributs** consisteix, doncs, a eliminar de la consideració les variables que tenen els seus pesos distribuïts uniformement entre les classes.

La mesura de la dispersió del pesos es calcula mitjançant la distància mitjana quadràtica entre el pesos de cada parell de classes. L'atribut amb la dispersió més baixa s'elimina.

Les distàncies es veuen amb més detall en el mòdul "Agregació (*clustering*)" d'aquesta assignatura.



Exemples de l'aplicació dels mètodes ID3 i LMD

En aquest gràfic podem veure els arbres resultants d'aplicar LMDT. Utilitzem el conjunt de valors sobre l'evolució postoperatòria de pacients d'un hospital.

Algun dels problemes que hem apuntat ara resideixen en la motivació pràctica dels mètodes de classificació propis de les xarxes neuronals que es veuen en el mòdul corresponent, i a altres tècniques com les *support vector machines* (Burges, 1998). 

6. Interpretació dels resultats obtinguts amb arbres de decisió

El resultat d'un procés de construcció d'un arbre de decisió és un classificador. El classificador ha induït una partició en el conjunt de dades originals. Cal veure què volen dir les particions.

En aquest sentit, i això és vàlid també per a altres models de classificació com les xarxes neuronals o les regles d'associació, cal saber de quina manera es divideixen les dades.

Normalment, les diverses eines de construcció d'arbres permeten de visualitzar alguns d'aquests aspectes:

a) Nombre d'observacions de cada classe per cada node i condició sobre valors: això dóna una idea del grau de puresa que hi ha a cada nivell de l'arbre.

b) Atribut més discriminant segons diversos criteris: normalment es valora segons la mateixa funció de construcció de l'arbre, però també s'acostumen a presentar ordenats segons el percentatge de separació de classes.

c) Resta d'atributs ordenats per valor de discriminació en aquell nivell: això permet de fer-se una idea de com és de bo cada atribut com a separador.

d) Taxa d'error a cada nivell de l'arbre.

e) Trajectòria de cada observació (com fa, per exemple, Sipina): això permet de conèixer l'estabilitat del procés de classificació, ja que podem comparar com una observació ha anat canviant de partició.

f) Matriu de confusió: mesura típica per a expressar la qualitat de les classes obtingudes amb un model (vegeu el mòdul sobre avaluació de models).

Lectures complementàries

Consulteu la visualització de les trajectòries d'una observació amb el programari Sipina en l'obra següent:

A. Zighed; J.P. Auray; G. Duru (1992). *Sipina: Méthode et Logiciel*. Lió: Editions A. Lacassagne.

Vegeu la matriu de confusió en el mòdul "Avaluació de models" d'aquesta assignatura.

7. Ponderació final dels arbres de decisió

En aquest apartat farem alguns comentaris finals amb relació als diversos temes de què hem tractat en aquest mòdul dedicat als arbres de decisió.

Preparació de dades

No tots els mètodes de construcció d'arbres admeten atributs amb valors continus. De fet, ID3 estava pensat per a funcionar amb atributs categòrics. En el cas de disposar de dades contínues, cal efectuar una discretització acurada que asseguri la mínima pèrdua d'informació. Si cal utilitzar un mètode que permeti la predicció de valors numèrics, llavors s'ha de recórrer als arbres que implementen regressió.

Avaluació d'arbres de decisió

El mètode típic per a avaluar un arbre de decisió és aplicar-lo a un conjunt de dades de prova i calcular el percentatge de casos classificats correctament o alguna altra mesura d'error de classificació de les que s'han comentat. Ja hem expressat de quina manera s'ha de ponderar la contribució de cada branca a l'error final de tot el model. Respecte a aquesta mesura bàsica, es pot seguir el mètode típic de validació creuada.

Vegeu la validació creuada en el mòdul "Avaluació de models" d'aquesta assignatura.

Poda

La poda simplifica el model sense que aquest perdi massa capacitat predictiva, però introdueix un cost addicional de càlcul.

Avantatges i inconvenients

Comentem algunes de les característiques principals dels arbres de decisió com a models:

1) Eficiència en classificació: el nombre de tests que s'han d'efectuar per a decidir si cal efectuar una partició o no és, en el pitjor dels casos, tan alt com el nombre d'atributs del domini.

2) Comprensibilitat: com més reduït és l'arbre que resulta, més senzilla n'és la interpretació. Amb arbres que consideren un gran nombre d'atributs i que generen un nombre de nivells alt, sorgeixen dificultats d'interpretació. En principi, doncs, cal preferir mètodes que donen arbres més "plans", bé perquè acumulen diversos tests en un mateix node (és a dir, tenen en compte més d'un atribut al mateix temps), bé perquè apliquen la poda. Per

contra, els arbres multivariants acostumen a tenir condicions més complexes en cada node.

3) Importància relativa dels diversos atributs per a la classificació: el nivell en què apareix un node dins l'arbre és una indicació de la rellevància de l'atribut corresponent per a la tasca de classificació i, per tant, ens dóna una interpretació addicional del domini.

4) Inconvenients: l'inconvenient principal pot residir en la dificultat de comprensió d'alguns mètodes que generen arbres massa complexos. D'altra banda, si el conjunt d'observacions defineix una sèrie de particions que no es poden separar linealment, els arbres que efectuen una divisió binària poden donar un mal rendiment. Igualment en el cas d'arbres multivariants que només generin superfícies (hiperplans) rectangulars.

Resum

Els arbres de decisió són un model de classificació que es basa en la idea de trobar atributs discriminants en el sentit que generen particions de classificació prou uniformes.

La formulació original dels arbres de decisió es pot remuntar fins als arbres basats en classificació i regressió proposats per Breiman, que, bàsicament, procedeixen iterativament, seleccionant a cada pas l'atribut més discriminant i dividint el conjunt de dades en dues particions segons el valor de tall escollit per a aquest atribut. La formulació més coneguda per a classificació és la de Quinlan i els seus mètodes d'inducció d'arbres de decisió descendent (*top-down*) i iterativa.

Una millora possible dels arbres de decisió és la incorporació de mètodes de poda per a impedir o retallar la generació de subarbres als quals corresponen particions amb un percentatge d'error en classificació no admissible.

Igual que passa amb altres sistemes de classificació, els arbres de decisió tenen problemes per a tractar dominis on no es pot assegurar la linealitat de les hipersuperfícies de separació entre les diverses classes. Una possible extensió i millora per a aquesta mena de problemes és la que presenten els arbres de decisió multivariants lineals d'Utgoff (LMDT, *linear multivariant decision trees*).

Finalment, els arbres de decisió admeten una traducció fàcil a regles de decisió.

Activitats

1. Accediu a l'adreça d'Internet que es dóna al marge i compareu les especificacions dels diversos programaris adreçats a construir arbres.

Per a fer l'activitat 1, accediu a l'adreça http://www.kdnuggets.com.

2. Per al problema que us havíeu proposat inicialment, us serveixen els arbres de decisió? Quin mètode creieu que us resultaria més convenient? Vegeu l'activitat 1 del mòdul "Extracció de coneixement a partir de dades" d'aquest curs.

3. Utilitzeu com a criteri de poda la taxa d'error esperada d'un arbre per a indicar quins nodes caldria podar amb el programari Sipina per al conjunt de dades i l'arbre que resulta d'aplicar el mètode ID3 següents. Es tracta d'una base de dades que podeu explorar més detalladament amb el programari Sipina. Recull dades de més de dos mil passatgers del *Titanic* que indiquen quina classe de passatge tenien (primera, segona, tercera classe i tripulació, que són 1, 2, 3 i 4, respectivament), l'edat (1 = 'Vell', 2 = 'Jove'), el sexe (1 = 'Home', 2 = 'Dona') i si van sobreviure o no (1 = 'Sí', 2 = 'No').

Passatger	Classe	Edat	Sexe	Sobrevivent
1	1	1	1	1
2	1	1	1	1
3	1	1	1	1
4	1	1	1	1
5	1	1	1	1
6	1	1	1	1
7	1	1	1	1
8	1	1	1	1
324	1	2	1	1
325	1	2	2	1
326	2	1	1	1
327	2	1	1	1
328	2	1	1	1
329	2	1	1	1
330	2	1	1	1
331	2	1	1	1
332	2	1	1	1
1.998	4	1	1	2
1.999	4	1	1	2
2.000	4	1	1	2
2.001	4	1	1	2
2002	4	1	1	2
2.003	4	1	1	2
2.004	4	1	1	2
2.005	4	1	1	2
2.006	4	1	1	2
2.007	4	1	1	2
9	1	1	1	1

4. Feu el mateix, amb el mètode CART.

Vegeu l'activitat 3 d'aquest mòdul.

Exercicis d'autoavaluació

Donat el **conjunt** següent de dades si l'atribut de classificació és *Risc*:

Sexe	Edat	Velocitat	Color cotxe	Risc
Dona	Vell	Ràpida	Vermell	No
Dona	Jove	Ràpida	Gris	Sí
Dona	Vell	Ràpida	Gris	No
Dona	Jove	Lenta	Vermell	Sí
Home	Jove	Lenta	Gris	No
Home	Jove	Lenta	Gris	No
Dona	Jove	Ràpida	Vermell	No
Home	Vell	Lenta	Vermell	No
Dona	Vell	Lenta	Vermell	No
Dona	Jove	Ràpida	Vermell	No
Home	Vell	Lenta	Vermell	No
Dona	Vell	Lenta	Vermell	No
Dona	Jove	Ràpida	Vermell	No
Dona	Jove	Lenta	Vermell	No
Dona	Vell	Ràpida	Gris	Sí
Home	Jove	Ràpida	Gris	Sí
Dona	Jove	Ràpida	Vermell	Sí
Dona	Vell	Ràpida	Vermell	No
Dona	Jove	Ràpida	Gris	Sí
Home	Jove	Lenta	Vermell	Sí
Dona	Vell	Ràpida	Vermell	No
Dona	Jove	Ràpida	Vermell	Sí
Dona	Jove	Ràpida	Vermell	No
Dona	Jove	Ràpida	Vermell	No
Home	Vell	Lenta	Vermell	No
Dona	Vell	Ràpida	Vermell	No
Dona	Jove	Ràpida	Vermell	No

a) Llisteu les particions que es poden obtenir atenent únicament al valor *Risc*.
b) Calculeu el valor d'informació (entropia) de cadascuna de les particions.

2. Apliqueu ID3 al conjunt de dades anterior suposant que l'atribut de classificació és *Risc*.

3. Trobeu la taxa d'error de l'arbre que heu obtingut respecte al conjunt de dades de prova següent.

Sexe	Edat	Velocitat	Color del cotxe	Risc
Dona	Jove	Ràpida	Gris	No
Home	Jove	Ràpida	Vermell	Sí
Home	Jove	Ràpida	Vermell	Sí

Sexe	Edat	Velocitat	Color del cotxe	Risc
Dona	Vell	Ràpida	Vermell	No
Home	Vell	Ràpida	Gris	Sí
Dona	Jove	Ràpida	Vermell	Sí
Home	Vell	Lenta	Vermell	No
Home	Jove	Ràpida	Gris	Sí
Dona	Vell	Ràpida	Vermell	Sí
Home	Vell	Lenta	Gris	No
Dona	Jove	Ràpida	Gris	No
Home	Jove	Ràpida	Vermell	No
Home	Vell	Lenta	Gris	Sí

Bibliografia

Breiman, L.; Friedman, H.; Olshen, R.; Stone, C. (1984). *Classification and Regression Trees*. Belmont: Wadsworth.

Brodley, C.E.; Utgoff, P. (1992). "Multivariate Decision Trees". *Technical Report, MASSCS-92-93*. Amherst: University of Massachusetts.

Burges, C.J. (1998). "A Tutorial on Support Vector Machines for Pattern Recognition". *Data Mining and Knowledge Discovery* (vol. 2, núm. 2).

Cendrowska, J. (1987). "PRISM: an Algorithm for Inducing Modular Rules". *International Journal of Man-Machine Studies* (vol 4, núm. 27, pàg. 349-370).

Esposito, F.; Malerba, D.; Semeraro, G. (1997). "A Comparative Analysis of Methods for Pruning Decision Trees". *IEEE Transactions on Pattern Analysis and Machine Intelligence* (vol. 19, pàg. 476-493).

Hartigan, J.A. (1975). *Clustering Algorithms*. Nova York: John Wiley & Sons.

Ileath, D.; Kasif, S.; Salzberg, S. (1993). "Induction of Oblique Decision Trees". *Proceedings of the 13th International Conference in Artificial Intelligence* (pàg. 1002-1007). IJCAI-93. San Francisco: Morgan Kaufmann.

Krichevsky, R.E.; Trofimov, V.K. (1983). "The Performance of Universal Coding". *IEEE Transactions on Information Theory* (vol. IT-27, núm. 2).

Lim, T.S.; Loh, W.Y.; Shih, Y.S. (1997). *An Empirical Comparison of Decision Trees and Other Classification Methods*. Technical Report TR-979. Madison: Department of Statistics, University of Wisconsin-Madison.

Lim, T.S.; Shih, Y.S. (1997). "Split Selection Methods for Classification Trees". *Statistica Sinica*.

López de Màntaras, R. (1991). "A Distance-Based Attribute Selection Measure for Decision Tree Induction". *Machine Learning* (vol. 6, núm. 1, pàg. 81-92).

Mingers, J. (1989). "An Empirical Comparison of Pruning Methods for Decision Trees Induction". *Machine Learning* (núm. 4, pàg. 227-243).

Quinlan, J.R. (1987). "Generating Production Rules from Induction Trees". *Proceedings of the Fourth International Machine Learning Workshop* (pàg. 304-307). San Francisco: Morgan Kaufmann Publishers.

Quinlan, J.R. (1987). "Simplifying Decisions Trees". *International Journal of Man-Machine Studies* (núm. 27, pàg. 221-234).

Quinlan, J.R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

Rissanen, M. (1985). "The Minimum Description Length Principle". A: S. Kotz; N.L. Johnson (ed.). *Encyclopedia of Statistical Sciences* (vol. 5). Nova York: John Wiley & Sons.

Rissanen, M. (1989). *Stochastic Complexity in Statistical Inquiry*. Nova Jersey World Scientific Publishers Company.

Zighed, A.; Auray, J.P.; Duru, G. (1992). *Sipina: Méthode et Logiciel*. Lió: Editions A. Lacassagne.