

4. Sailkapen-zuhaitzak

Irakasgaia: Konputazio-Zientzien Metodo Matematikoak

Titulazioa: Informatikan Ingeniaria

Konputazio Zientzia eta Adimen Artifiziala saila

Universidad del País Vasco - Euskal Herriko Unibertsitatea

Aurkibidea

- 1 TDIDT oinarrizko algoritmoa
- 2 ID3 algoritmoa (Quinlan, 1986)
- 3 C4.5 algoritmoa (Quinlan, 1993)

Aurkibidea

- 1 **TDIDT oinarrizko algoritmoa**
- 2 ID3 algoritmoa (Quinlan, 1986)
- 3 C4.5 algoritmoa (Quinlan, 1993)

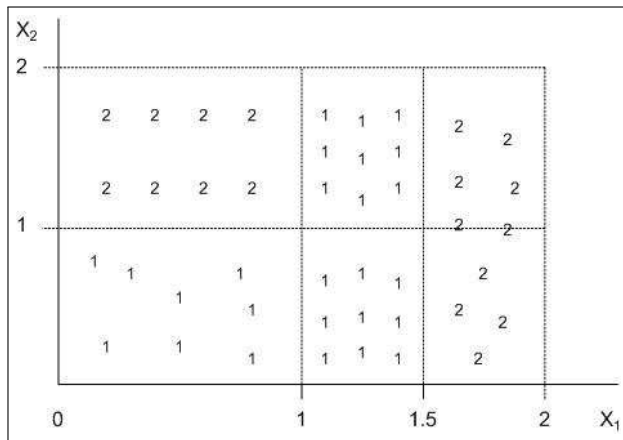
Sarrera

- Sailkapen-zuhaitz bat egitura hierarkikoan antolatutako **baldintza-multzo bat** da. Zuhaitzaren errotik hasi eta hostoetako batera iritsi arte betetzen diren baldintzei jarraituz hartuko da azken erabakia.
- Sailkapen-zuhaitz bat **induzitzea**, modu errekurtsiboan aldagai iragarleen definizio-eremua **partizio disjuntuetan** zatitzean datza.
- Partizio bat erregelen multzo bat da non **elkarrekiko disjuntuak** diren eta guztien artean **definizio-eremu osoa** hartzen duten.
- Indukzioaren ataza honakoan datza: entrenamendurako kasuen multzoan oinarrituz (training set) sailkapenerako erregela bat garatu behar da edozein kasuren atributuak aztertuz bere klasea iragarriko duena

Sarrera

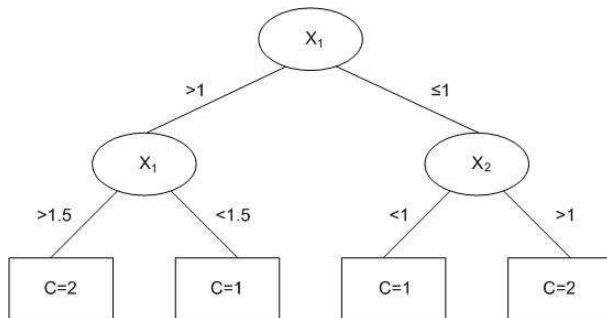
- Atributuek nahiko informazio ematen al dute indukzioaren ataza burutu ahal izateko?
- Atributu **egokiak** erabiliz, test-multzoko kasuak ondo sailkatuko dituen erabaki-zuhaitz bat eraikitzea da helburua
- Normalean, erabaki-zuhaitz asko aurki daitezke
- Erabaki-zuhaitzen indukziorako algoritmoak honakoetan bereizten dira nagusiki:
 - Partizioen aukeraketarako irizpideak
 - Inausketarako estrategiak

Oinarritzko algoritmoa



X_1 eta X_2 aldagai iragarleen bidez eta bi balio posible dituen C klase-aldagaiaren bidez karakterizatutako kasuen adierazpena planoan

Oinarrizko algoritmoa



Aurreko adibideari dagokion sailkapen-zuhaitza

Sailkapen-zuhaitzarekin baliokide den erregela-multzoa:

R_1 :	If	$X_1 > 1.5$		then	$C = 2$
R_2 :	If	$1 < X_1 < 1.5$		then	$C = 1$
R_3 :	If	$X_1 < 1$	and	$X_2 < 1$	then $C = 1$
R_4 :	If	$X_1 < 1$	and	$X_2 > 1$	then $C = 2$

Oinarritzko algoritmoaren sasikodea:

TDIDT (Top Down Induction of Decision Trees)

Input: Etiketaturako N kasuen D multzoa, horietako bakoitza n aldagai iragarleren bidez $X_1, \dots, X_n$, eta C klase-aldagaiaren bidez karakterizatuta dagoena

Output: Sailkapen-zuhaitza

Begin **TDIDT (Top Down Induction of Decision Trees)**

- if D multzoko kasu guztiak c klase berekoak badira
- then
 - induzkioaren emaitza erpin simple bat da (hostoa), c etiketaduna
- else
 - begin
 1. Informazio handieneko X_r aldagaia aukeratu, $x_r^1, \dots, x_r^{n_r}$ balioak dituen
 2. X_r aldagaiaren n_r balioen arabera D -ren partizioa egin: $D_1, \dots, D_{n_r}$ sortu
 3. $D_1, \dots, D_{n_r}$ horietarako n_r azpizuhaitz $T_1, \dots, T_{n_r}$ eraiki
 4. X_r eta n_r azpizuhaitzak $T_1, \dots, T_{n_r}$ lotu $x_r^1, \dots, x_r^{n_r}$ balioen bidez
 - end

endif

End TDIDT

Sailkapen-zuhaitzen inausketa

- Zergatik inausi? Nahikoa informazio ematen ez duten aldagai iragarleak baztertzeke, induzitutako zuhaitza handiegia ez izateko (ez litzateke interpretagarria)
- Bi estrategia daude: **prepruning** (informazio-teoria eta errore kopurua ebaluatzeko neurri estatistikoak erabiliz, adar gehiago ez garatzea erabaki daiteke) eta **postpruning** (adarrak garatu ondoren inaustea erabaki daiteke)

Aurkibidea

- 1 TDIDT oinarrizko algoritmoa
- 2 ID3 algoritmoa (Quinlan, 1986)
- 3 C4.5 algoritmoa (Quinlan, 1993)

ID3 algoritmoa

- Informazio handieneko aldagaiaren aukeraketa aldagai iragarleen eta klase-aldagaiaren arteko elkarrekiko-informazio kantitatean oinarritzen da (**informazio-irabazia** edo **information gain**):

$$I(X_i, C) = H(C) - H(C|X_i)$$

- Matematikoki froga daiteke irizpide horrek **balio-kopuru handieneko aldagaien aukeraketari lehentasuna** ematen diela
- Aldez aurretiko aldagai-aukeraketa egiten da, X_i aldagai iragarle bakoitzaren eta C klase-aldagaiaren arteko independentzia-test batean oinarritua (**prepruning**)

Aurkibidea

- 1 TDIDT oinarrizko algoritmoa
- 2 ID3 algoritmoa (Quinlan, 1986)
- 3 C4.5 algoritmoa (Quinlan, 1993)**

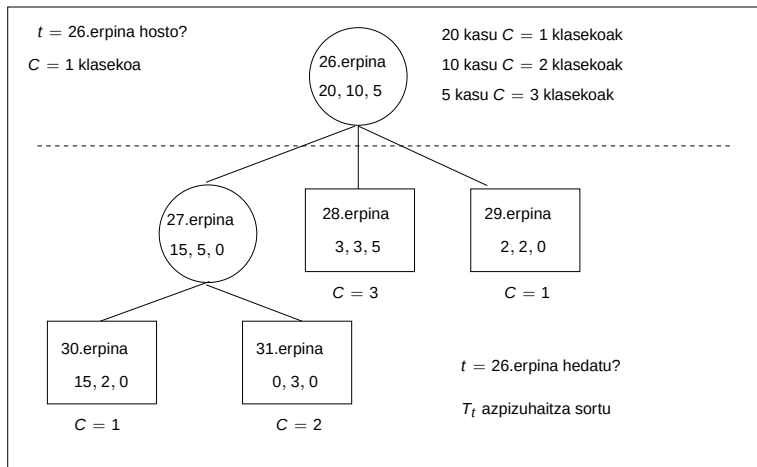
C4.5 algoritmoa

- C4.5 algoritmoan (Quinlan, 1993), honela definitzen den **irabazi-ratio** edo **gain ratio** irizpidean oinarritzen da informazio handieneko aldagaiaren aukeraketa

$$\frac{I(X_i, C)}{H(X_i)}$$

- Matematikoki froga daiteke **balio posible gehieneko aldagaiei ez zaiela lehentasunik ematen** aldagaien aukeraketan
- Sailkapen-zuhaitza induzitu eta gero **zuhaitzaren inausketa** egiten da, hipotesi-test batean oinarrituta; adar bat hedatu izanak merezi ote duen erabaki behar da (**postpruning**)

C4.5 algoritmoaren postpruning estrategia



Irudia: C4.5 algoritmoaren inausketa prozesurako adibidea

C4.5 algoritmoa (postpruning inausketa)

Zuhaitzaren inausketa prozesua

- $N(t) = 35$, $t = 26$ erpineko kasu-kopurua
- $e(t) = 10 + 5 = 15$, t erpinean gaizki sailkatuta dagoen kasu-kopurua
- $n'(t) = e(t) + \frac{1}{2} = 15,5$, $e(t)$ ren jarraitasunezko zuzenketa
- T_t , t erpinetik abiatuz hedatuko den azpizuhaitza
- $h(T_t) = 4$, T_t azpizuhaitzaren hosto-kopurua

$$\bullet \quad n'(T_t) = \sum_{i=1}^{h(T_t)} e(i) + \frac{h(T_t)}{2} = 2 + 0 + 6 + 2 + \frac{4}{2} = 12$$

T_t azpizuhaitzaren hostoetan dagoen errore-kopurua

$$\bullet \quad S(n'(T_t)) = \sqrt{\frac{n'(T_t)[N(t) - n'(T_t)]}{N(t)}} = \sqrt{\frac{12(35 - 12)}{35}} \simeq 2.8$$

$n'(T_t)$ ren desbideratzea

t erpina hedatu egingo da $\Leftrightarrow n'(T_t) + S(n'(T_t)) < n'(t) \Leftrightarrow 12 + 2.8 < 15.5$
 26 erpina hedatu egingo da, 28 , 29 , 30 eta 31 erpinak sortuz

Bibliografia

- Quinlan J.R. (1986) *Induction of decision trees*. Machine Learning, 1(1), 81-106
- Quinlan J.R. (1993) *C4.5: Programs for Machine Learning*. Morgan Kaufmann
- Introducción a la Minería de Datos. Capítulo 11: Árboles de decisión y sistemas de reglas. J.Hernandez Orallo et al.
- http://en.wikipedia.org/wiki/Decision_tree_learning
- http://en.wikipedia.org/wiki/ID3_algorithm
- http://en.wikipedia.org/wiki/C4.5_algorithm
- http://en.wikipedia.org/wiki/Decision-tree_pruning
- <http://en.wikipedia.org/wiki/Overfitting>
- Overfitting, 9min slidecast:
<http://blog.lokad.com/journal/2009/4/22/overfitting-when-accuracy-measure-goes-wrong.html>