4. Sailkapen-zuhaitzak

Irakasgaia: Konputazio-Zientzien Metodo Matematikoak
Titulazioa: Informatikan Ingeniaria
Konputazio Zientzia eta Adimen Artifiziala saila
Universidad del País Vasco–Euskal Herriko Unibertsitatea

4.1 Sarrera

Gai honetan *sailkapen-zuhaitzak* edo *erabaki-zuhaitzak* izeneko paradigma aurkeztuko da. Paradigma honetan aldagai iragarleen definizio-eremuaren *partiketa errekurtsiboa* eginez, problemari buruzko ezagutza zuhaitz-egituran adierazi ahal izango da.

4.2 atalean ereduaren indukziorako oinarrizko algoritmoaren ideia nagusienak aurkeztuko dira. Ondoren, 4.3 eta 4.4 ataletan $ID3$ eta $C4.5$ algoritmoak azalduko dira.

4.2 Oinarrizko algoritmoa

Atal honetan TDIDT (*Top Down Induction of Decision Trees*) algoritmoaren oinarrizko ideiak azalduko ditugu; etiketatutako kasuen datu-multzo batetik abiatuz sailkapen-zuhaitza induzitzeko algoritmo gehienak jasotzen dituela esan daiteke.

1. Irudian TDIDT algoritmoaren sasikodea ikus daiteke. Algoritmoaren ideia nagusia

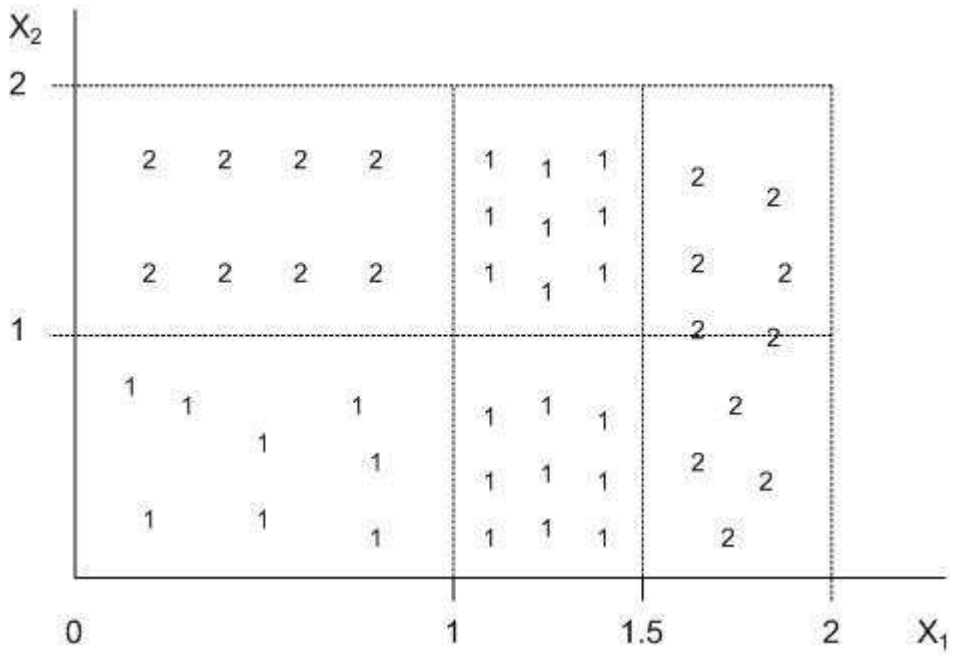
---

Sarrera: Etiketatutako N kasuen D multzoa, horietako bakoitza n aldagai iragarleen bidez karakterizatuta dagoena, $X_1, \ldots, X_n$, eta C klase-aldagaia
Irteera: Sailkapen-zuhaitza
Hasiera TDIDT
  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, $x_r^1, \ldots, x_r^{n_r}$ balioak dituena
      2. X_r aldagaiaren n_r balioen arabera D multzoaren partizioa egin: $D_1, \ldots, D_{n_r}$ sortu
      3. $D_1, \ldots, D_{n_r}$ horietarako n_r azpizuhaitz $T_1, \ldots, T_{n_r}$ eraiki
      4. X_r eta n_r azpizuhaitzak $T_1, \ldots, T_{n_r}$ lotu $x_r^1, \ldots, x_r^{n_r}$ balioen bidez
    end
  endif
End TDIDT

Irudia 1: TDIDT algoritmoaren sasikodea

honakoa da: sailkapen-zuhaitzaren adar bati dagozkion kasu guztiak klase berekoak ez diren bitartean, adar horretan aukeratua izan ez den aldagaien artetik bat aukeratuko da, informazio gehien ematen duena edo aurretik ezarritako irizpide bati jarraituz egokiena dena. Aldagai horren aukeraketari esker zuhaitza hedatu egingo da aldagaiak har ditzakeen balio posibleak adina adarretan.

2. Irudian grafikoki adierazten dira X_1 eta X_2 aldagai iragarleen bidez eta bi balio posible (1 eta 2) dituen C klase-aldagaiaren bidez karakterizatutako kasuak. 2. Iru-



Irudia 2: X_1 eta X_2 aldagai iragarleen bidez eta bi balio posible dituen C klase-aldagaiaren bidez karakterizatutako kasuen adierazpena planoan

dian erakutsitako adibidearen ezagutza adierazteko modu bat 3. Irudiko sailkapen-zuhaitzaren bidez ikus daiteke.

3. Irudian ikus daitekeen bezala, aldagai iragarleak borobil baten barruan adieraziak izango dira sailkapen-zuhaitzean; zuhaitzaren hostoak aldiz laukien bidez adieraziak izango dira, bertan idatziko delarik sailkapen-zuhaitzak adar horietatik jaisten diren kasuei egokituko dien klase-aldagaiaren balioa.

3. Irudiko sailkapen-zuhaitzaren adar guztien *sakonera* 2-koa da, sakoneraren kontzeptuak sailkapen-zuhaitzaren *konplexutasuna*ri buruzko ideia bat emango digularik.

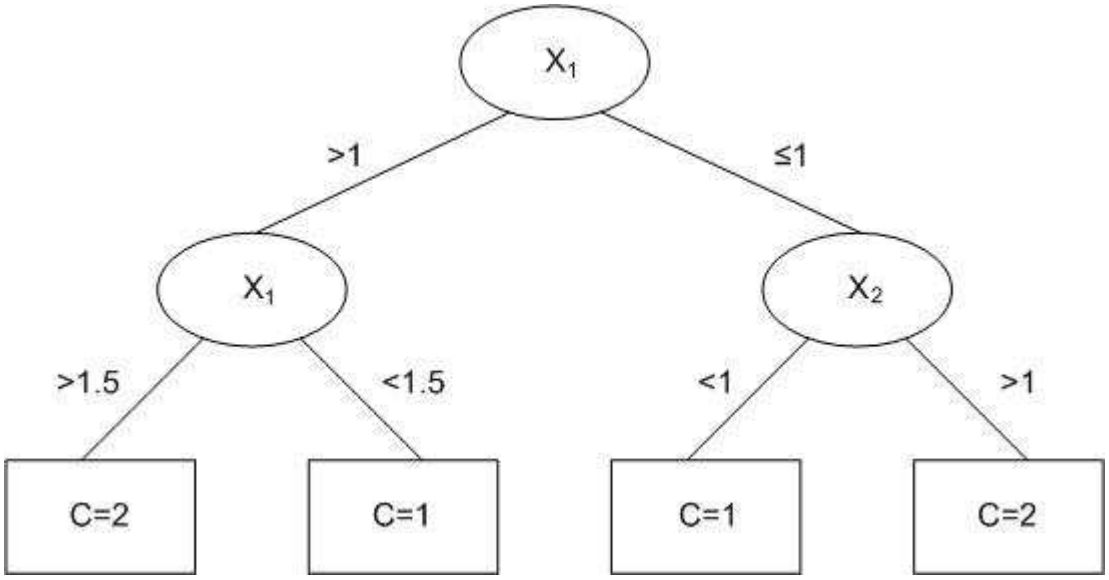
Bestalde, adibide horretan *zarata*rik ez dagoela esan dezakegu, hau da hostoak *puruak* direla, klase-aldagaiarekiko balio berbereko kasuak biltzen dituztelako. Esan behar da egoera hori ez dela arrunta gertatzen problema erreal bat eredutzerakoan.

Amaitzeko, 3. Irudiko sailkapen-zuhaitza erregela-multzo baten bidez adieraziko dugu.

Honako R_1, R_2, R_3 eta R_4 erregelak eta 3. Irudiko sailkapen-zuhaitza baliokideak dira.

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

Ohartarazi behar da, erregela-multzo batek ez duela beti bere baliokidea den sailkapen-zuhaitz bat. Horregatik esaten da erregelen indukzioaren paradigma sailkapen-zuhaitzena baino malguagoa dela.



Irudia 3: 2. Irudian adierazitako adibideari dagokion sailkapen-zuhaitza

4.3 $ID3$ algoritmoa

Sailkapen-zuhaitzen indukziorako algoritmo ezagunenetako bat $ID3$ algoritmoa da, Quinlanek 1986an sortua. Algoritmo horretan informazio gehieneko aldagaia aukeratzeko erabiltzen den irizpidea aldagai horren eta klase-aldagaiaren arteko *elkarrekiko informazioan* edo *mutual information* kontzeptuan oinarritzen da. Kontestu honetan $I(X_i, C)$ elkarrekiko informazioari *informazio-irabazia* edo *information gain* esaten zaio.

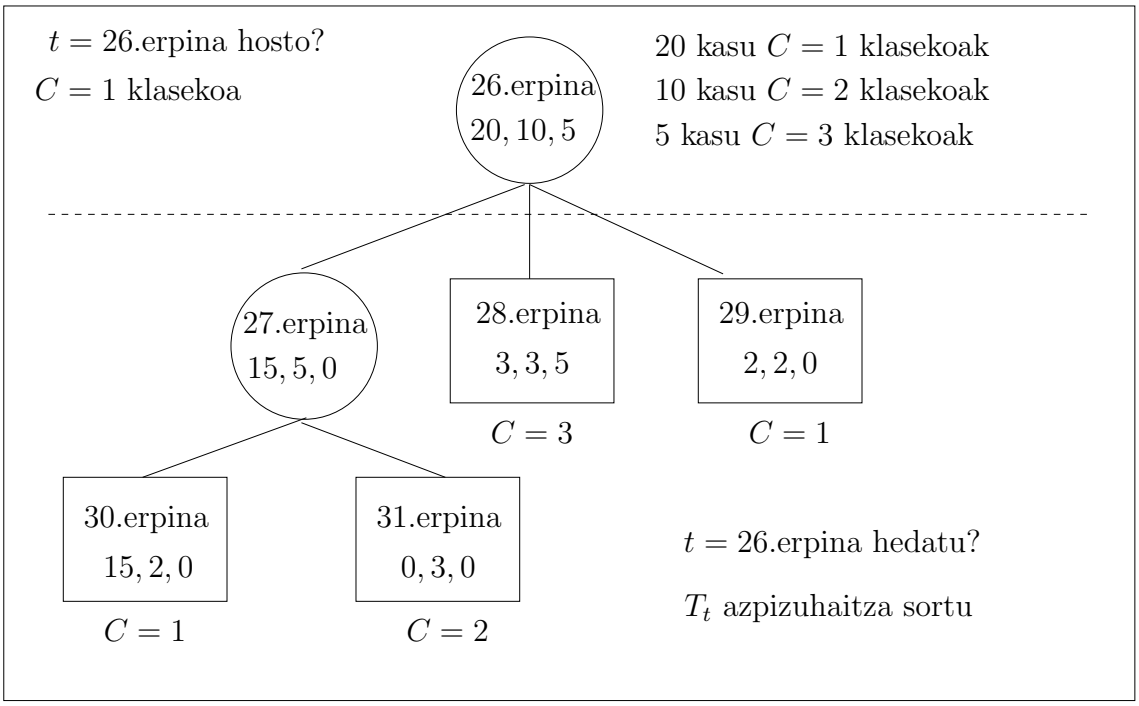
$I(X_i, C) = H(C) - H(C|X_i)$ betetzen denez, X_i eta C arteko elkarrekiko informazioak adierazten duena zera da: X_i aldagaiaren balioa ezagutzeagatik C aldagaiari buruzko ziurgabetasuna zenbat murrizten den.

Matematikoki froga daiteke $ID3$ algoritmoak aldagaia aukeratzeko erabiltzen duen irizpide hau ez dela bidezkoa balio-kopuru handieneko aldagaiei lehentasuna ematen baitie.

Gainera, $ID3$ algoritmoak aldez aurretiko aldagai-aukeraketa bat egiten du, kontestu honetan *prepruning* izenez ezagutzen dena. Horretarako, X_i aldagai iragarle bakoitzaren eta C klase-aldagaiaren arteko independentzia-test bat egiten da. Sailkapen-zuhaitzaren indukzio-prozesuan independentzia-hipotesi testa baztertzen duten aldagai iragarleak besterik ez dira kontuan izango.

4.4 $C4.5$ algoritmoa

Quinlan-ek 1993 urtean $ID3$ algoritmoaren hobekuntza bat eman zuen, $C4.5$ izenekoa. $C4.5$ algoritmoa $I(X_i, C)/H(X_i)$ moduan definitzen den *irabazi-ratio* edo *gain ratio* irizpidearen erabilpenean oinarritzen da. Modu horretan, balio posible gehieneko aldagaiei aukeraketan lehentasunik ez ematea lortzen da. Gainera, $C4.5$ algoritmoak sailkapen-zuhaitza induzitu eta gero *zuhaitzaren inausketa* egiten du. Inausketa hipotesi-test batean oinarritzen da, non erabaki behar den ea adar bat hedatzeak merezi ote duen.



Irudia 4: $C4.5$ algoritmoaren inausketa prozesurako adibidea

Har dezagun 4. Irudiko azpizuhaitza. Jakin nahi duguna da, zuhaitzeko 26. erpina hedatzea komeni den, hedatzearen ondorioz 28, 29, 30 eta 31 erpinak sortzeko. 26 erpina ez hedatzea erabakiko bagenu 15 akats egingo genituzke; hedatzea erabakiz gero aldiz, 10 akats besterik ez genuke egingo. Hortaz, egokiena erpina hedatzea dela badirudi ere, induzitutako sailkapen-zuhaitzaren erabilgarritasuna modu orokortuan ikusi behar da.

Testa burutu ahal izateko honako terminoak kontsideratuko ditugu:

- $N(t)$: t. erpina hedatu bai ala ez erabakitzeko testa egitean t. erpin horretan dagoen adibide-kopurua. Adibidean $t = 26$ erpina hedatu bai ala ez erabakitze-prozesuan gaude, eta $N(t) = 35$ adibide daude bertan.
- $e(t)$: t. erpinean dagoen gaizki sailkatutako adibide-kopurua. Adibidean $e(t) = 10 + 5 = 15$ adibide daude gaizki sailkatuta.
- $n'(t)$: $e(t)$ren zuzenketa, $e(t)$ri $\frac{1}{2}$ gehituz lortzen dena, hau da, $n'(t) = e(t) + \frac{1}{2}$.

Aurreko hiru terminoek t erpinari erreferentzia egiten badiote, hurrengoak t erpinetik abiatuz hedatuko den T_t azpizuhaitzarekin erlazionatuta daude.

- $h(T_t)$: T_t azpizuhaitzak duen hosto-kopurua da. Adibidean $h(T_t) = 4$ hosto daude.
- $n'(T_t)$: T_t azpizuhaitzaren hostoetan dagoen errore-kopurutik abiatuz lortzen da, eta honela definituta dago:

$$n'(T_t) = \sum_{i=1}^{h(T_t)} e(i) + \frac{h(T_t)}{2}$$

Adibidean $n'(T_t) = 2 + 0 + 6 + 2 + \frac{4}{2} = 12$ da.

- $S(n'(T_t))$: $n'(T_t)$ren desbideratzea, honako formularen arabera definitua:

$$S(n'(T_t)) = \sqrt{\frac{n'(T_t)[N(t) - n'(T_t)]}{N(t)}}$$

Adibidean $S(n'(T_t)) = \sqrt{\frac{12(35-12)}{35}} \approxeq 2.8$

t erpina hedatu eta T_t azpizuhaitza sortzearen erabakia honako erregelari jarraituz hartzen da:

t erpina hedatuko da $\Leftrightarrow n'(T_t) + S(n'(T_t)) < n'(t)$.

Adibidean $n'(T_t) + S(n'(T_t)) = 12 + 2.8 < 15.5 = n'(t)$ denez, 26 erpina hedatu eta 28, 29, 30 eta 31 erpinak sortuko dira.

Erreferentziak

1. Breiman L., Friedman J.H., Olshen R. A. , Stone P. J. (1984) *Classification and Regression Trees*. Wadsworth International Group.
2. Quinlan J. R. (1986) Induction of decision trees. *Machine Learning*, 1(1), 81–106.
3. Quinlan (1993) *C4.5: Programs for Machine Learning* Morgan Kaufmann.