OSNOVE UMETNE INTELIGENCE 2018/19

učenje dreves iz šumnih podatkov manjkajoči atributi

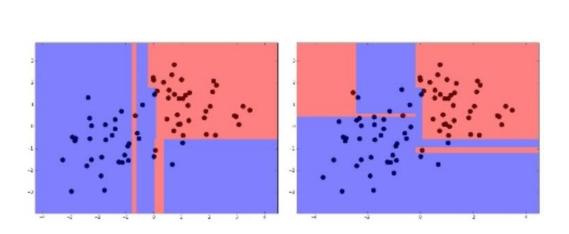
Pregled

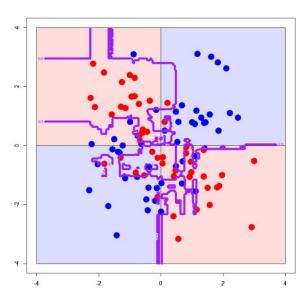


- strojno učenje
 - uvod v strojno učenje
 - učenje odločitvenih dreves
 - učenje dreves iz šumnih podatkov (rezanje dreves)
 - manjkajoči atributi

Učenje dreves iz šumnih podatkov

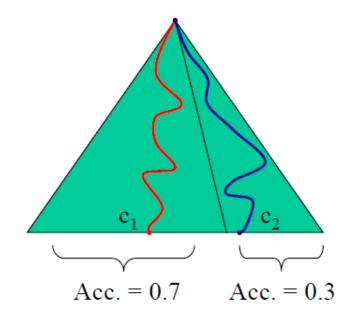
- vir: nepopolni podatki, napake v učnih primerih
- težave:
 - učenje šuma in ne dejanske (skrite) funkcije, ki generira podatke
 - pretirano prilagajanje vodi v velika drevesa
 - slaba razumljivost dreves
 - posledica: nižja klasifikacijska točnost na novih podatkih
- rešitev: rezanje odločitvenega drevesa





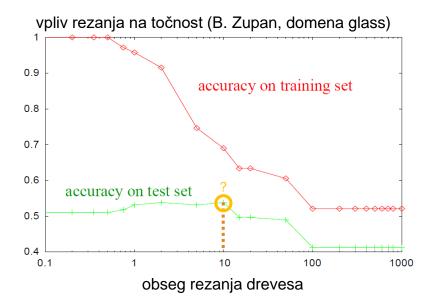
Rezanje odločitvenih dreves

- premislek: nižji deli drevesa (bližji listom) predstavljajo večje lokalno prilagajanje učnim podatkom, ki so lahko posledica šuma
- ideja: odstranimo (režemo) spodnje dele drevesa, da dosežemo boljšo posplošitev naučenega drevesa (in klasifikacijsko točnost na nevidenih podatkih)
- primer nizke točnosti drevesa pri skrajnem primeru pretiranega prilagajanja:
 - dva razreda, c_1 in c_2 , $p(c_1) = 0.7$, $p(c_2) = 0.3$
 - privzeta točnost (točnost večinskega razreda) = 0,7
 - drevo, zgrajeno do konca (en primer v vsakem listu)
 - pričakovana točnost: $0.7 \times 0.7 + 0.3 \times 0.3 = 0.58$ (manj kot privzeta točnost!)



Rezanje odločitvenih dreves

- cilj: maksimiziraj pričakovano točnost (minimiziraj pričakovano napako) drevesa
- vprašanja:
 - kako to doseči,
 - kje rezati,
 - kombinatorično število možnih porezanih dreves
- primeri:

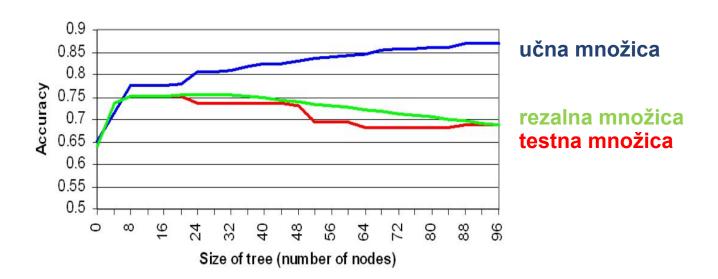


primer iz prakse: lociranje primarnega tumorja (domena *Primary tumor*

	Klas. točnost
Pretirano pril. drevo (150 vozlišč)	41%
Porezano drevo (15 vozlišč)	45%
Privzeta točnost	24,7%
Zdravniki	42%

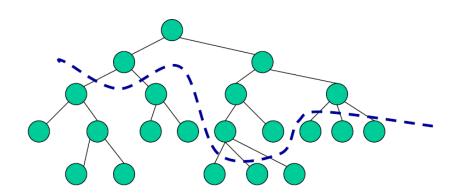
Rezalna množica

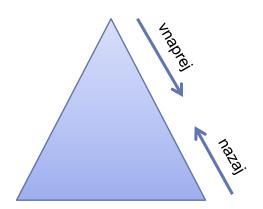
- ocenjevanje točnosti poddrevesa pri rezanju:
 - na učnih podatkih
 - na posebni množici testnih primerov (rezalna množica, validacijska množica) če imamo dovolj podatkov (ostane manj podatkov za gradnjo)
- tipična delitev podatkov
 - učna množica (70%): od tega množica za gradnjo (growing set) 70% in rezalna množica (pruning set) 30%
 - testna množica (30%)



Strategije rezanja

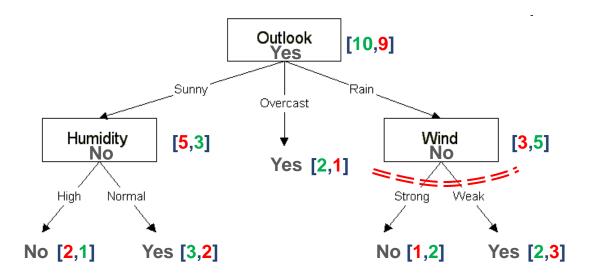
- **rezanje vnaprej** (angl. *forward pruning, pre-stopping*): uporaba dodatnega kriterija za zaustavitev gradnje grevesa glede na obseg šuma (na podlagi: števila primerov, večinski razred, smiselnost delitve v poddrevesa glede na informacijski prispevek itd.)
 - hitrejše
 - kratkovidno, upošteva samo zgornji del drevesa
- rezanje nazaj (angl. post-pruning): rezanje, ki po gradnji celotnega drevesa, odstrani manj zanesljive dele drevesa (opisujejo šum, zgrajeni iz manj podatkov in z manj informativnimi atributi)
 - počasneje, oblika post-procesiranja
 - upošteva informacijo iz celega drevesa
 - pristopa:
 - rezanje z zmanjševanjem napake (reduced error pruning, REP)
 - rezanje z minimizacijo napake (minimal error pruning, MEP)

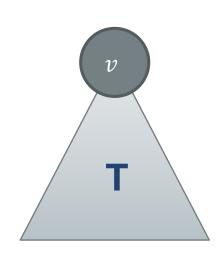




Rezanje z zmanjševanjem napake

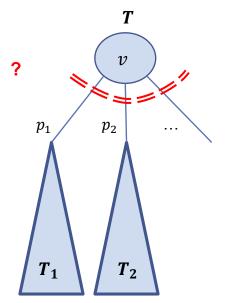
- angl. reduced error pruning (REP)
- uporablja rezalno množico, potrebna primerna velikost za zanesljivost
- postopek:
 - potuj od listov navzgor (prični s starši listov)
 - za vsako notranje vozlišče v izračunaj dobitek rezanja: št. napačnih klasifikacij v drevesu T — št. napačnih klasifikacij v vozlišču v
 - če je dobitek pozitiven, obreži in nadaljuj postopek s staršem sicer ustavi postopek





Rezanje z minimizacijo napake

- angl. minimal error pruning (MEP) (Niblett in Bratko, 1986; Cestnik in Bratko, 1991)
- uporablja množico za gradnjo drevesa (in ne ločene rezalne množice)
- cilj: poreži drevo tako, da je ocenjena klasifikacijska točnost maksimalna (napaka minimalna)
- za vozlišče v izračunamo:
 - **statično napako** (verjetnost klasifikacije v napačen razred) $e(v) = p(razred \neq C|v)$, C je večinski razred v v
 - vzvratno napako (angl. backed-up error) $\sum_i p_i E(T_i) = p_1 E(T_1) + p_2 E(T_2) + \cdots$
- režemo, če je statična napaka manjša od vzvratne napake
- napaka optimalno obrezanega drevesa je torej $E(T) = \min(e(v), \sum_i p_i E(T_i))$ E(T) = e(v), če je v list



Ocenjevanje verjetnosti

- kako oceniti statično napako v vozlišču v?
- primeri uporabe relativne frekvence (N št. primerov v vozlišču, n št. primerov, ki pripadajo večinskemu razredu C):
 - $N = 1, n = 1 \rightarrow \text{točnost} = 100\%$
 - $N = 2, n = 1 \rightarrow \text{točnost} = 50\%$? (samo z enim dodatnim primerom)

težave:

- potrebujemo oceno verjetnosti, ki je stabilna tudi pri manjšem številu primerov
- smiselno je, da ocena verjetnosti upošteva tudi apriorno verjetnost (verjetnost, ki jo poznamo o problemu npr. 50% za izid meta kovanca)



Ocenjevanje verjetnosti

boljši oceni verjetnosti:

Laplaceova ocena verjetnosti:

$$p = \frac{n+1}{N+k}$$

n – št. primerov, ki pripadajo razredu C,

N − št. vseh primerov

k – št. vseh razredov

• k je problematičen parameter; ocena ne upošteva apriorne verjetnosti

· m-ocena verjetnosti

delež upoštevanja apriorne verjetnosti relativne frekvence

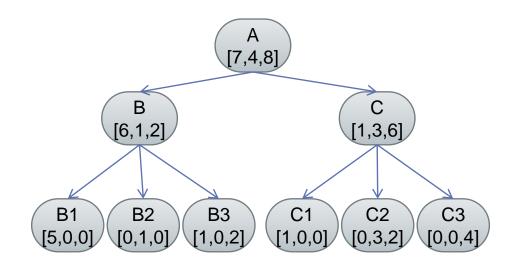
$$p = \frac{n + p_a m}{N + m} = p_a \cdot \frac{m}{N + m} + \frac{n}{N} \cdot \frac{N}{N + m}$$

 $p_a\,$ – apriorna verjetnost razreda C

m – parameter ocene (vpliva na delež upoštevanja apriorne verjetnosti)

- malo šuma majhen m malo rezanja / veliko šuma velikm veliko rezanja
- posplošitev Laplaceove ocene za m = k in $p_a = 1/k$

- primer: Bratko: Prolog Programming for Al
- Podano je odločitveno drevo za klasifikacijo v tri razrede (x, y in z) z naslednjimi apriornimi verjetnostmi razredov: $p_a(x) = 0.4$, $p_a(y) = 0.3$, $p_a(z) = 0.3$. Številke v oglatih oklepajih [x, y, z] predstavljajo frekvence primerov v vozlišču, ki pripadajo ustreznim razredom. Obreži drevo s postopkom MEP in vrednostjo m = 8.



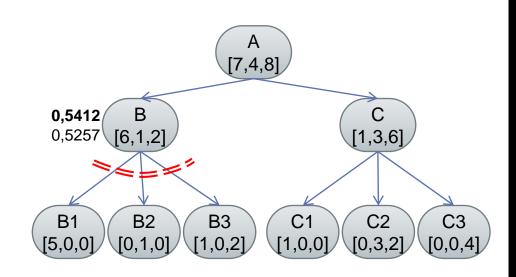
klasifikacijske točnosti v listih B1, B2 in B3:

$$p(x|B1) = \frac{n+m \cdot p_a(x)}{N+m} = \frac{5+8 \cdot 0.4}{5+8} = 0,6308$$
$$p(y|B2) = \frac{1+8 \cdot 0.3}{1+8} = 0,3778$$
$$p(z|B3) = \frac{2+8 \cdot 0.3}{3+8} = 0,4$$

- vzvratna točnost v vozlišču B: $\frac{5}{9} \cdot 0.6308 + \frac{1}{9} \cdot 0.3778 + \frac{3}{9} \cdot 0.4 = 0.5257$
- statična točnost v vozlišču B:

$$p(x|B) = \frac{6+8\cdot0.4}{9+8} = 0.5412$$

- statična točnost je večja od vzvratne točnosti → porežemo
- nadaljujemo z vozliščema
 C in A ...



klasifikacijske točnosti v listih C1, C2 in C3:

$$p(x|C1) = \frac{n+m \cdot p_a(x)}{N+m} = \frac{1+8 \cdot 0.4}{1+8} = 0.4667$$

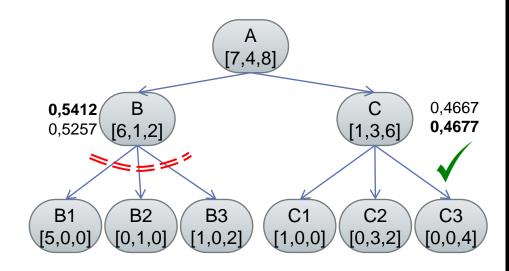
$$p(y|C2) = \frac{3+8 \cdot 0.3}{5+8} = 0.4154$$

$$p(z|C3) = \frac{4+8 \cdot 0.3}{4+8} = 0.5333$$

- vzvratna točnost v vozlišču C: $\frac{1}{10} \cdot 0,4667 + \frac{5}{10} \cdot 0,4154 + \frac{4}{10} \cdot 0,5444 = 0,4677$
- statična točnost v vozlišču C:

$$p(z|C) = \frac{6+8\cdot0.3}{10+8} = 0.4667$$

- vzvratna točnost je večja od statične točnosti ne porežemo
- nadaljujemo z vozliščem A ...



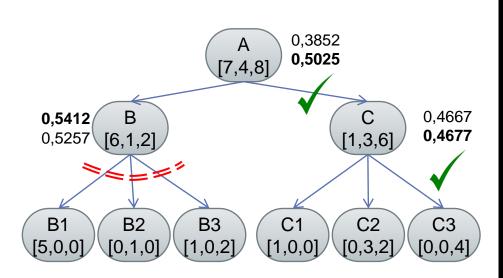
klasifikacijske točnosti v podrevesih s koreni v B in C:

$$E(B) = \min(e(B), \sum_{i} p_{i}E(B_{i})) = 0.5412$$

 $E(C) = \min(e(C), \sum_{i} p_{i}E(C_{i})) = 0.4677$

- vzvratna točnost v vozlišču A: $\frac{9}{19} \cdot 0.5412 + \frac{10}{19} \cdot 0.4677 = 0.5025$
- statična točnost v vozlišču A: $p(z|A) = \frac{8+8\cdot0.3}{19+8} = 0.3852$

statične točnosti → ne porežemo



Pregled



- strojno učenje
 - uvod v strojno učenje
 - učenje odločitvenih dreves
 - učenje dreves iz šumnih podatkov (rezanje dreves)
 - manjkajoči atributi

Obravnava atributov

potrebno je nasloviti še naslednja problema:

- manjkajoči podatki v atributih:
 - ignorirati cele primere z neznanimi vrednostmi?
 - uporabiti vrednost NA/UNKNOWN?
 - nadomestiti manjkajočo vrednost (povprečna, najbolj pogosta, naključna, napovedana)
 - primer obravnavamo verjetnostno glede na vse možne vrednosti atributa (s tako utežjo lahko sodeluje pri gradnji modela in klasifikaciji)

	Variables			
Respondent	Α	В	С	D
1	1	2	3	4
2	1	2	3	4
3	4	3	2	1
4	4	3	2	1
5	1	2		1
6		2	2	1
7	1	2	2	
8	1		2	1

- obravnava numeričnih atributov: običajno izvedemo diskretizacijo v dva (binarizacija) ali več diskretnih intervalov
 - intervali z enako frekvenco primerov (equal-frequency)
 - intervali enake širine (equal-width)
 - intervali, ki maksimizirajo informacijski dobitek

