Izvod iz **Praktikuma za vežbe iz Veštačke inteligencije, L.Stoimenov, A.Milosavljević, Elektronski fakultet u Nišu, 2004**

7.5. ID3: indukovanje stabala odluke

ID3 [Quinlan 1986] je algoritam za učenje na osnovu sličnosti u primerima, baziran na indukciji. Radi se o algoritmu opšte namene, tj. ne postoji znanje o domenu ugrađeno u strategiju traženja. Celokupno znanje o domenu je ugrađeno kroz jezike za predstavljanje primera i koncepata, koji se inicijalizuju na početku svake sesije. Primeri se zadaju u obliku vektora nominalnih atributa i ceo skup primera mora biti poznat na početku učenja. Algoritam je nehijerarhijski, utoliko što uči mali broj koncepata na istom nivou apstrakcije i novi koncepti se ne mogu učiti na osnovu prethodno naučenih. Ne postoje slučajni elementi u procesu formiranja koncepta, tako da će rezultati biti isti pri svakoj eksploataciji algoritma.

Koncepti koje ovaj algoritam formira su taksonomskog tipa, logički reprezentovani u obliku stabla odluke. Koren stabla predstavlja atribut koga treba prvo testirati i u zavisnosti od njegove vrednosti kreće se dalje po stablu. Svaki list u stablu predstavlja jedan od koncepata, a svaki unutrašnji čvor predstavlja test koji se vrši nad primerom da bi se klasifikovao. Tako se problem učenja svodi na traženje ključnih atributa, koji mogu klasifikovati primere. Stablo se generiše od vrha prema dnu, polazeći od celog skupa primera i mere entropije. Originalna verzija ID3 algoritma omogućavala je klasifikovanje primera u dve klase, ali je proširenje algoritma za rad sa više klasa jednostavno.

Količina informacija koju dobijemo od događaja E obrnuto je proporcionalna verovatnoći njegovog ostvarivanja p(E):

$$I(E) = \log_2 (1 / p(E)) = - \log_2 p(E)$$

Entropija (ili iznenađenje) je prosečna količina informacija koju nosi *n* različitih događaja, odnosno prosečna količina informacija potrebna za generisanje klase primera. Ako događaj predstavlja pripadnost određenoj klasi (konceptu), i ako postoji *n* klasa $\{C_1, C_2, ... C_n\}$, entropija će biti:

$$H = - \text{Sum}_{j=1,n} (p(C_j) * \log_2 p(C_j))$$

gde je $p(C_j)$ verovatnoća da proizvoljno izabran primer pripada konceptu (klasi) C_j. Ovako određena entropija je konstantna za jedan skup primera.

Očekivana količina informacija potrebnih za formiranje stabla čiji je koren atribut A sa vrednostima $\{A_1, A_2, ... , A_m\}$ je:

$$E(A) = \text{Sum}_{i=1,m} (p(A_i)*H(A_i))$$

gde je $p(A_i)$ verovatnoća da proizvoljno izabrani primer ima A_i kao vrednost atributa A, a $H(A_i)$ entropija podskupa primera koji imaju A_i kao vrednost atributa A. Razlika između entropije za slučaj kada nije poznata vrednost atributa i očekivane količine informacija u slučaju poznate vrednosti atributa predstavlja dobitak u informacijama kada se posmatrani atribut koristi kao kriterijum za razvrstavanje primera.

$$G(A) = H – E (A)$$

ID3 određuje atribut sa najvećim dobitkom, tj. preferira atribut koji nosi najviše informacija za ceo skup primera. Kako je H konstantno za skup primera (ne zavisi od izabranog atributa), to ID3 zapravo preferira atribut sa minimalnom očekivanom količinom informacija. Iz čvora koji je obeležen izabranim atributom postoji onoliko grana koliko vrednosti ima izabrani atribut i originalni skup primera deli se u disjunktne podskupove prema vrednostima tog atributa. Proces se ponavlja rekurzivno, sve dok svi primeri u posmatranom podskupu ne pripadaju istoj klasi.

ID3 algoritam (za dve klase):

1. Ako svi primeri pripadaju istoj klasi, kreiraj list.
2. Inace, nađi najbolji atribut A, dodaj granu za svaku vrednost atributa A, rasporedi primere u podskupove.
3. Ako su svi primeri klasifikovani tacno, stop.
4. Inace, primeni korake 1-3 za listove.

7.6. Rad ID3 algoritma

Razmotrimo rad algoritma ID3 na primeru iz [Quinlan, 1986]. Primeri su opisani pomoću četiri atributa:

- oblačnost, sa vrednostima sunčano, oblačno, kiša,
- temperatura, sa vrednostima toplo, sveže, hladno,
- vlažnost, sa vrednostima normalna, visoka, i
- vetar, sa vrednostima true, false.

Neka postoje dve klase, P i N.

Lista primera predstavljena u Lisp notaciji je:

((N (suncano toplo visoka false)) (N (suncano toplo visoka true))

(P (oblacno toplo visoka false)) (P (kisa sveze visoka false))

(P (kisa hladno normalna false)) (N (kisa hladno normalna true))

(P (oblacno hladno normalna true)) (N (suncano sveze visoka false))

(P (suncano hladno normalna false)) (P (kisa sveze normalna false))

(P (suncano sveze normalna true)) (P (oblacno sveze visoka true))

(P (oblacno toplo normalna false)) (N (kisa sveze visoka true)))

Ako sa *p* označimo broj primera koji pripadaju klasi P, a sa *n* broj primera koji pripadaju klasi N, entropija se može odrediti prema formuli:

$$H(p,n) = - (p/(p+n))* \log_2 (p/(p+n)) – (n/(p+n)) * \log_2 (n/(p+n))$$

Kako je p=9, a n=5, to je H = 0.940 bitova. Posmatrajmo prvi atribut vreme. Pet od 14 primera ima prvu vrednost ovog atributa (sunčano), od kojih p_1=2 pripadaju klasi P, a n_1=3 klasi N. Entropija ovog podskupa primera iznosi $H(p_1, n_1)$=0.971 bitova. Za preostale vrednosti atributa vreme (oblačno i kiša) imamo:

p_2=4, n_2=0, $H(p_2, n_2) = 0$

p_3=3, n_3=2, $H(p_3, n_3) = 0.971$

Očekivana količina informacija za atribut vreme je:

$E(\text{vreme}) = (5/14)* H(p_1, n_1) + (4/14)* H(p_2, n_2) + (5/14)* H(p_3, n_3)$

Dobitak u informacijama je:

$G(\text{vreme}) = H - E(\text{vreme}) = 0.246$ bitova.

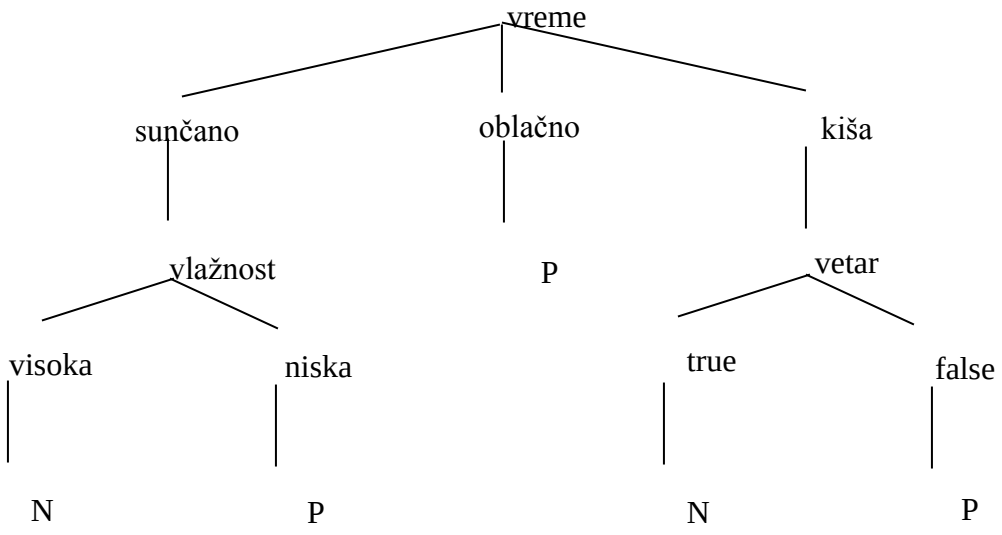
Na sličan način dobija se:

G(temperatura) = 0.029 bitova,

G(vlažnost) = 0.151 bitova,

G(vetar) = 0.048 bitova,

Kako atribut vreme daje najveći dobitak, to se on bira za koren stabla. Skup primera se deli na podskupove primera koji sadrže odgovarajuće vrednosti ovog atributa. Svi primeri koji imaju oblačno za vrednost atributa vreme pripadaju klasi P, te se formira list sa tom oznakom. Za ostale podskupove ponavlja se isti postupak. Na taj način dobija se sledeće stablo odluke:



U slučaju kada postoji izuzetno veliki broj primera, koristi se iterativni postupak. Formira se stablo za jedan podskup primera (prozor) i ono se dalje testira za preostale primere. U slučaju pogrešnog klasifikovanja nekog primera, postupak formiranja stabla se ponavlja za početni skup uvećan tim primerom, sve dok se svi primeri ne klasifikuju ispravno. Proces nije osetljiv na veličinu početnog skupa, a vreme obrade linearno raste sa složenošću procesa. Originalni algoritam daje veoma komplikovana stabla u slučajevima kada primeri sadrže šum, ali postoje njegova proširenja koja ovaj problem prevazilaze.

Stabla odluke nemaju izražajne moći logike predikata prvog reda, ali i pored toga omogućavaju rešavanje nekih praktičnih problema, kao što je indukovanje pravila za ekspertne sisteme u nekoliko medicinskih domena, prepoznavanje slika, proučavanje pozicija u šahu i sličnih.