Mašinsko učenje

Stabla odlučivanja

Nemanja Maček

Sadržaj

- Uvodne napomene
- Osnovni algoritmi učenja stabala obuke
- Traženje najboljeg atributa za klasifikaciju
- Učenje stabala odluke u prostoru hipoteza
- Induktivni pomeraj
- Praktični problemi učenja stabala odluke
- Zaključne napomene

Uvodne napomene

Šta su stabla odlučivanja?

- Stabla odlučivanja se sastoje od čvorova i grana, koje spajaju čvorove roditelje sa čvorovima naslednika – dece.
- Čvor koji se nalazi na vrhu stabla (koren) nema roditelja, dok svi ostali tzv. unutrašnji čvorovi imaju samo jednog roditelja.
- Čvorovi koji nemaju naslednike se nazivaju listovi; listovi predstavljaju sva moguća rešenja koja se mogu dobiti iz datog stabla.
- To su čvorovi odgovora, a ostali se nazivaju čvorovi odluke.

Šta su stabla odlučivanja?

- Stabla odluke klasifikuju primere u dve ili više klasa na osnovu vrednosti atributa kojima su primeri opisani, propuštajuci ih niz stablo od korena ka listovima.
- Na početku se bira atribut čija vrednost najbolje deli raspoložive primere.
- Svaki čvor u stablu predstavlja test nekog atributa, a svaka grana koja polazi iz tog čvora odgovara jednoj od mogućih vrednosti tog atributa.
- Na taj način se primeri dele u podskupove, zavisno od vrednosti izabranog atributa.
    - Ako svi atributi u podskupu pripadaju istoj klasi, stablu se dodaje list.
- Svaki put kroz stablo predstavlja jedno klasifikaciono pravilo, tj. konjunkciju testova atributa, a samo stablo je disjunkcija ovih konjunkcija.

Uvodne napomene

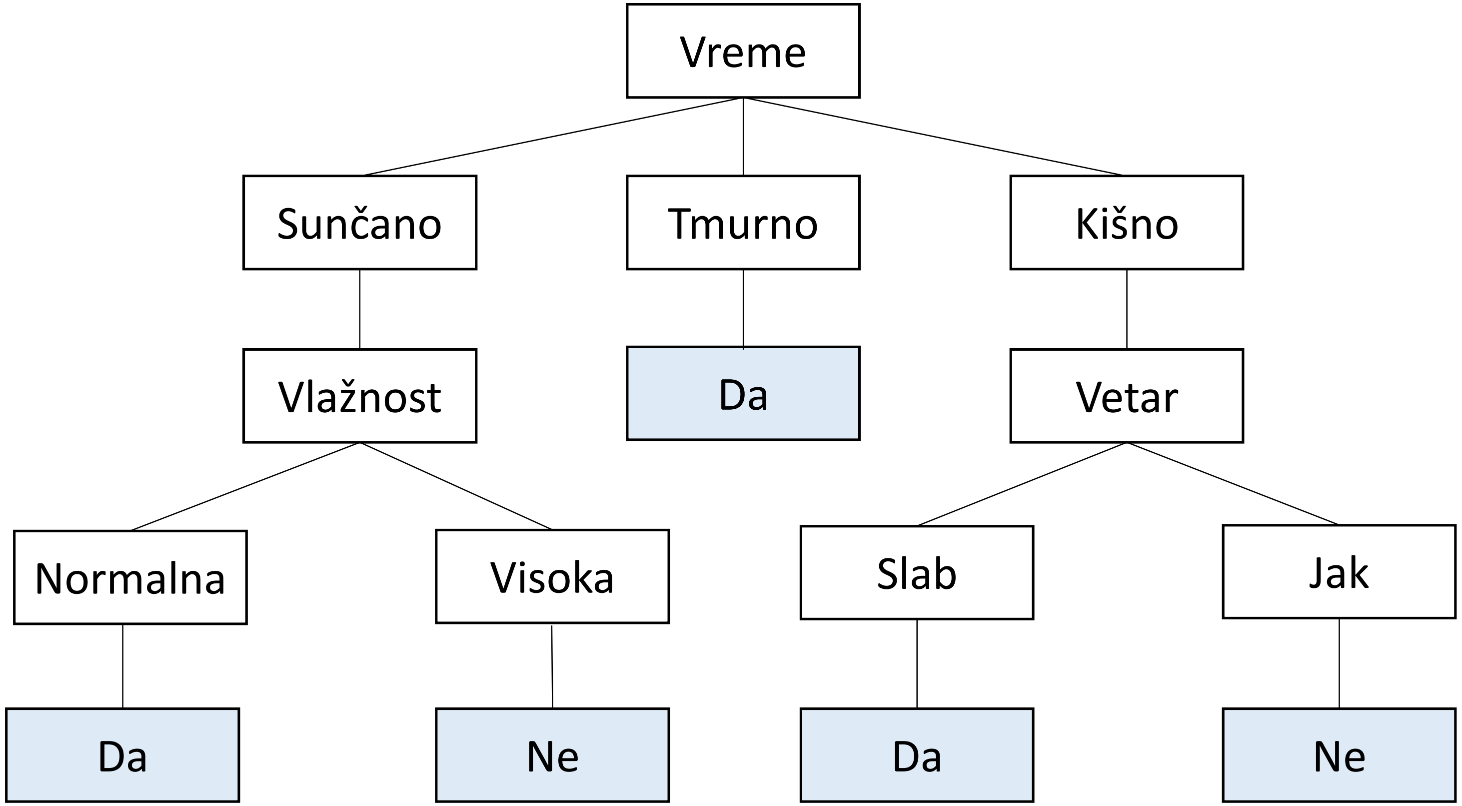
Šta su stabla odlučivanja?

- Stabla odluke su značajna za oblast mašinskog učenja, jer su zbog svoje sistematske strukture razumljiva za ljude i prestavljaju odličnu tehnologiju obrazlaganja u automatizovanim dijagnostičkim sistemima.
- Obrazloženje zašto je doneta neka konkretna odluka se dobija izdvajanjem puta od pripadajućeg lista do korena, pri čemu se iz svakog čvora iščitava razlog parcijalnog izbora.
- Osim toga, svakom stablu odlučivanja se jednoznačno može pripisati skup pravila tipa: ako – onda (if – then), koja su osnovni gradivni blokovi baza znanja ekspertskih i drugih sistema zasnovanih na znanju.

Jednostavan primer.

- Na str. 7 prikazan je jedan primer stabla odluke.
- Ovo stablo klasifikuje subotnja jutra na osnovu toga da li je vreme pogodno za igranje tenisa ili ne.
- Čvorovima odgovaraju atributi (vreme, vlažnost, vetar), dok listovima odgovaraju klasifikacije (Da – vreme pogodno za igranje tenisa i Ne – vreme nije pogodno za igranje tenisa).

Jednostavan primer.



Jednostavan primer.

- Ono što stabla odlučivanja čini tehnikom široko prihvaćenom u praksi je i njihova veza sa pravilima odlučivanja koja dominiraju u simboličkom pristupi veštačkoj inteligenciji.
- Stablu sa str. 8 odgovaraju ekvivalentna pravila vezana za listove, koje ćemo numerisati sa leva na desno:
- List 1: **Ako** je: Vreme: Sunčano i Vlažnost: Normalna **Onda**: IgratiTenis Da
- List 2: **Ako** je: Vreme: Sunčano i Vlažnost: Visoka **Onda**: IgratiTenis Ne
- List 3: **Ako** je: Vreme: Tmurno **Onda**: IgratiTenis Da
- List 4: **Ako** je: Vreme: Kišno i Vetar: Slab **Onda**: IgratiTenis Da
- List 5: **Ako** je: Vreme: Kišno i Vetar: Jak Onda: IgratiTenis Ne.

Uvodne napomene

Dimenzije razlikovanja stabala obuke.

- Postoji više dimenzija na osnovu kojih se stabla odluke mogu razlikovati:
  - Testovi mogu biti multivarijabilni (testiraju više atributa ulaznih oblika odjednom) ili monovarijabilni (testiraju samo jedan atribut).
  - Testovi mogu imati dva ili više ishoda. Ako svi testovi u stablu odluke imaju po dva ishoda, onda je to binarno stablo odluke.
  - Atributi mogu biti kategorički ili numerički. Binarni atributi se formalno mogu trezirati i kao numerički i kao kategorički.
  - Možemo imati dve ili više klasa. Ako imamo dve klase i binarne ulaze, stablo reprezentuje jednu Bulovu funkciju i naziva se Bulovo stablo odluke.

Uvodne napomene

Kada su stabla odlučivanja pogodna za upotrebu?

- Primeri su predstavljeni vektorima, tj. <u>parovima atribut-vrednost</u>. Opisani su fiksnim skupom atributa, npr. *Temperatura*, i njihovim vrednostima, npr. *Vruće*.
- Za stabla odlučivanja je najjednostavnija situacija kada svaki atribut obuhvata <u>mali broj disjunktnih vrednosti</u> (npr. *Vruće*, *Toplo*, *Hladno*), ali je dozvoljeno i da atributi imaju realne vrednosti, npr. numeričko predstavljanje atributa *Temperatura*.
- Ciljne funkcije imaju <u>diskretne izlazne vrednosti</u>. U slučaju kontinualnih izlaznih vrednosti moguće je izvršiti <u>transformaciju u diskretne</u> (nominalne) promenljive kroz proces diskretizacije. Vrednosti ciljnjih funkcija su linearne logičke kombinacije vrednosti atributa.

Kada su stabla odlučivanja pogodna za upotrebu?

- Podaci za obuku mogu da sadrže greške – šum. Metodi učenja stabala odluke su robusni na greške u klasifikaciji obučavajućih primera, kao i na greške u vrednostima atributa koje opisuju ove primere.
- Obučavajućim podacima mogu da nedostaju vrednosti atributa. Metodi stabala odluke mogu da se koriste čak i kada neki obučavajući primeri imaju nepoznate vrednosti, npr. ako je *Vlažnost* određenog dana poznata samo za neke obučavajuće primere.

Osnovni algoritmi učenja stabala obuke

Osnovni algoritam.

- Većina algoritama koji su razvijeni za učenje stabala odluke su varijacije bazičnog algoritma koji koristi od-vrha-naniže (engl. *top-down*) pohlepnu (engl. *greedy*) pretragu kroz prostor mogućih stabala odluke.
- Ovaj pristup je korišcen kod ID3 algoritma i njegovog naslednika C4.5.

Osnovni algoritmi učenja stabala obuke

Osnovni algoritam.

- Osnovni algoritam ID3 uči stabla odluke konstruišući ih od vrha naniže, počevši pitanjem: koji atribut treba testirati u korenu stabla?
  - Da bi se odgovorilo na ovo pitanje, atribut svakog primera se procenjuje pomoću statističkog testa, da bi se odredilo koliko dobro klasifikuju obučavajuće uzorke.
- Odabere se najbolji atribut i koristi kao test u korenu stabla.
- Potom se generišu čvorovi naslednici za svaku moguću vrednost ovog atributa, a obučavajući primeri se propuštaju kroz grane i sortiraju na dostignutim čvorovima naslednicima.
  - Proces odredjivanja najboljeg atributa za testiranje u svakom čvoru nasledniku se ponavlja, koristeći obučavajuće primere koji su dospeli do tog čvora.
- Na taj nacin se ustari vrši pohlepna potraga za odgovarajućim stablom odluke.
- Shodno svojstvu pohlepne pretrage, algoritam se nikada ne vraća u prethodno stanje da bi eventualno preispitao ranije izbore.

Traženje najboljeg atributa za klasifikaciju

Kvantitativna mera vrednosti atributa.

- Glavni izbor u ID3 algoritmu je određivanje atributa koji se testirati u svakom čvoru stabla.
- Stoga nam je potrebna kvantitativna mera koja govori o vrednosti atributa.
- Definišemo statističku veličinu, informacionu dobit, koja meri koliko dobro može dati atribut da razdvoji obučavajuće primere na osnovu njihove ciljne klasifikacije.
- ID3 koristi veličinu informacioni dobitak da među kandidatima odabere najbolji atribut u svakom koraku dok stablo raste.

Entropija.

- Da bismo precizno definisali informacioni dobitak, podsetimo se smisla i definicije entropije kao mere informacija i neodredjenosti jednog sistema.
- Za datu klasu S, koja sadrži pozitivne (p) i negativne (n) primere nekog ciljnog koncepta, entropija od S u odnosu na ovu binarnu klasifikaciju je:

$$Entropy(S) = Entropy([p,n]) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$

- gde je $p_+ = p/(p+n)$ proporcija pozitivnih primera u S, a $p_- = n/(p+n)$ proporcija negativnih primera u S.

Traženje najboljeg atributa za klasifikaciju

Entropija.

- Pretpostavimo da je S klasa od 14 primera nekog binarnog koncepta i sadrži $p = 9$ pozitivnih i $n = 5$ negativnih primera.
- Usvajamo notaciju [9+, 5-].
- Tada je entropija u odnosu na ovu binarnu klasifikaciju:

$$Entropy([9+, 5-]) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0{,}940$$

- Entropija je 0 ako svi članovi S pripadaju istoj klasi, bilo pozitivnoj ili negativnoj.
- Entropija je 1 ako pozitivna i negativna klasa imaju isti broj primera.

Traženje najboljeg atributa za klasifikaciju

Entropija.

- Do sada smo diskutovali o entropiji u specijalnom slučaju kada je ciljna klasifikacija binarna.
- Generalno, ako ciljni atribut može uzeti jednu od c razlicitih vrednosti, onda entropiju definišemo kao:

$$Entropy(S) = \sum_{i=1}^{c} -p_i \log_2 p_i$$

- gde je p_i proporcija primera u S koji pripadaju klasi i.

Traženje najboljeg atributa za klasifikaciju

Informaciona dobit.

- Pošto je entropija mera nehomogenosti u klasama obučavajućih primera, sada možemo da definišemo meru efektivnosti atributa pri klasifikaciji obučavajućih primera.
- Mera koju ćemo koristiti, informacioni dobitak (engl. *information gain*), je <u>očekivana redukcija entropije</u> koju izaziva particija primera na osnovu ovog atributa.
- Preciznije, informacioni dobitak za slučaj da se atribut A koristi za razvrstavanje primera iz S, tj. kao koren stabla, je:

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy\,(S_v)$$

- gde je gde je $Values(S)$ skup svih mogucih vrednosti atributa A, a S_v podskup od S za koji atribut A ima vrednost v, tj. $S_v = \{s \in S | A(S) = v\}$.

Traženje najboljeg atributa za klasifikaciju

Informaciona dobit.

- Prvi izraz na desnoj strani jednačine je entropija originalne klase primera S, a drugi izraz je očekivana vrednost entropije nakon što je S podeljeno uz pomoć atributa A.
- $Gain(S, A)$ je u stvari informacija o vrednosti ciljne funkcije za poznatu vrednost atributa A.

Traženje najboljeg atributa za klasifikaciju

Sažeti ID3 algoritam.

- Sažeti ID3 algoritam specijalizovan za funkcije sa binarnim vrednostima atributa.
- Ovaj proces se nastavlja sve dok stablo ne klasifikuje primere idealno, ili dok ne iskoristi sve atribute.
- *ID3(Primeri, Ciljni_atribut, Atributi)*
    - *Primeri* su obučavajući primeri.
    - *Ciljni_atribut* je atribut čiju vrednost stablo odluke treba da predvidi.
    - *Atributi* je lista ostalih atributa koje može da testira stablo odluke.
    - Vraća stablo odluke koje korektno klasifkuje *Primere*.

Traženje najboljeg atributa za klasifikaciju

- Kreiraj čvor Koren
- Ako su svi *Primeri* pozitivni, vrati stablo *Koren* sa jednim cvorom i *oznaka* = +
- Ako su svi *Primeri* negativni, vrati stablo *Koren* sa jednim cvorom i oznaka = -
- Ako je *Atributi* prazno, vrati stablo *Koren* sa jednim cvorom i *oznaka* = najčešca vrednost *Ciljni_atribut* u *Primeri*
- Inače **počni**
    - $A \leftarrow$ atribut iz *Atributi* koji najbolje* klasifikuje *Primeri*
    - Atribut odluke za *Koren* $\leftarrow A$
    - Za svaku mogucu vrednost v_i od A
        - Dodaj novu granu stablu ispod *Koren*, koja odgovara testu $A=v_i$
        - Neka *Primeri* v_i bude podskup od *Primeri* koji za A ima vrednost v_i
        - Ako je *Primeri* v_i prazno
            - Onda ispod ove nove grane dodaj list i *oznaka* = najcešca vrednost *Ciljni_atributi* u *Primeri*
            - Inače ispod ove nove grane dodaj podstablo: **ID3(*Primeri* v_i, *Ciljni_atribut*, Atributi_{A])**
- **Kraj**
- Vrati Koren

*Najbolji atribut je onaj koji ima najveci informacioni dobitak

Primer.

- Pretpostavimo da je S skup obučavajućih primera – dana, opisanih atributima među kojima je i *Vetar* (uzima vrednosti *Slab* ili *Jak*).
- Kao i ranije, pretpostavimo da je S klasa od 14 primera, [9+,5-].
- Od ovih 14 primera, pretpostavimo da je kod 6 pozitivnih i 2 negativna *Vetar=Slab*, a kod ostalih je *Vetar=Jak*.
- Informacioni dobitak na osnovu sortiranja originalnih 14 primera po atributu *Vetar* se može izračunati na način prikazan na str 23.

Traženje najboljeg atributa za klasifikaciju

Primer.

$$Vrednosti(Vetar) = Slab, Jak$$
$$S = [9+, 5-]$$
$$S_{Slab} \leftarrow [6+, 2-]$$
$$S_{Jak} \leftarrow [3+, 3-]$$

$$Gain(S, Vetar) = Entropy(S) - \sum_{v \in \{Slab, Jak\}} \frac{|S_v|}{|S|} Entropy\,(S_v)$$

$$Gain(S, Vetar) = Entropy(S) - \frac{8}{14} Entropy(S_{Slab}) = -\frac{6}{14} Entropy\left(S_{Jak}\right)$$

$$Gain(S, Vetar) = 0{,}940 - \frac{8}{14} \times 0{,}811 - \frac{6}{14} \times 1{,}00 = 0{,}048$$

Traženje najboljeg atributa za klasifikaciju

Primer.

- Informacioni dobitak je veličina koju ID3 algoritam koristi da odredi najbolji atribut u rastućem stablu.
- Informacioni dobitak dva razlicita atributa *Vlažnost* i *Vetar* se računa da bismo odredili koji od njih je bolji za klasifikaciju obučavajućih primera datih na str 25.
- *Vlažnost* pruža veci informacioni dobitak od *Vetar* u odnosu na ciljnu klasifikaciju.
- Za datu klasu S od 9 pozitivnih i 5 negativnih primera, nakon sortiranja po atributu *Vlažnost* dobijamo klase [3+,4-] (*Vlažnost*=*Visoka*) i [6+,1-] (*Vlažnost*=*Normalna*).
- Informacioni dobitak prilikom ovakve podele je 0,151, u odnosu na informacioni dobitak kod atributa *Vetar* od samo 0,048.

Traženje najboljeg atributa za klasifikaciju

Obučavajući skup.

- Dan, Vreme, Temperatura, Vlažnost, Vetar, IgratiTenis
- D1, Suncano, Vruce, Visoka, Slab, Ne
- D2, Suncano, Vruce, Visoka, Jak, Ne
- D3, Tmurno, Vruce, Visoka, Slab, Da
- D4, Kišno, Toplo, Visoka, Slab, Da
- D5, Kišno, Hladno, Normalna, Slab, Da
- D6, Kišno, Hladno, Normalna, Jak, Ne
- D7, Tmurno, Hladno, Normalna, Jak, Da
- D8, Suncano, Toplo, Visoka, Slab, Ne
- D9, Suncano, Hladno, Normalna, Slab, Da
- D10, Kišno, Toplo, Normalna, Slab, Da
- D11, Suncano, Toplo, Normalna, Jak, Da
- D12, Tmurno, Toplo, Visoka, Jak, Da
- D13, Tmurno, Vruce, Normalna, Slab, Da
- D14, Kišno, Toplo, Visoka, Jak, Ne

Primer.

- Pogledajmo prvi korak algoritma, u kome se kreira najviši čvor stabla.
- Postavlja se pitanje koji je to atribut koji treba da testiramo prvi u stablu odluke.
- ID3 određuje informacioni dobitak za svaki kandidovani atribut, tj. za *Vreme*, *Temperatura*, *Vlažnost* i *Vetar*, potom selektuje onaj sa najvećim informacionim dobitkom.
- Vrednosti informacionog dobitka za sve atribute iznose:
  - $Gain(S, Vreme) = 0{,}246$
  - $Gain(S, Vlažnost) = 0{,}151$
  - $Gain(S, Vetar) = 0{,}048$
  - $Gain(S, Temperatura) = 0{,}029$
- gde S predstavlja klasu obučavajućih primera sa str. 25.

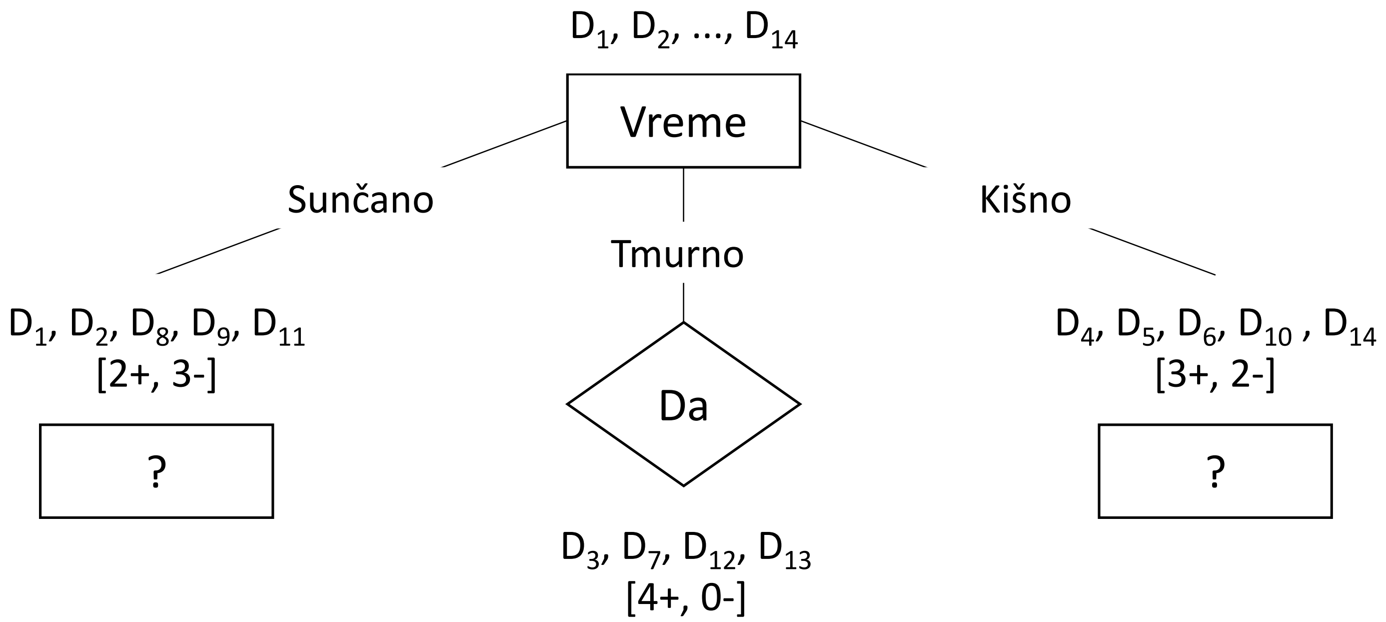
Traženje najboljeg atributa za klasifikaciju

Primer.

- Atribut *Vreme* za zadate obučavajuće primere daje najbolju predikciju ciljnog atributa *IgratiTenis*.
- Stoga je *Vreme* selektovano kao odlučujuci atribut za koren, a grane se postavljaju ispod korena za svaku od njegovih mogucih vrednosti, tj. *Sunčano*, *Tmurno* i *Kišno*.
- Rezultujuće <u>nedovršeno stablo odluke</u> je dato na str. 28 zajedno sa obučavajućim primerima sortiranim u svakom sledećem novom čvoru.
- Primetimo da je svaki primer za koji je *Vreme=Tmurno* takođe pozitivan primer za *IgratiTenis*. Stoga ovaj čvor stabla <u>postaje list</u> sa klasifikacijom *IgratiTenis=Da*.
- Nasuprot tome, sledujući čvorovi od *Vreme=Sunčano* i *Vreme=Kišno* još uvek <u>nemaju nultu entropiju</u>, usled čega stablo odluke mora i dalje da se razvija ispod ovih čvorova.

Traženje najboljeg atributa za klasifikaciju

Primer – nedovršeno stablo obuke.



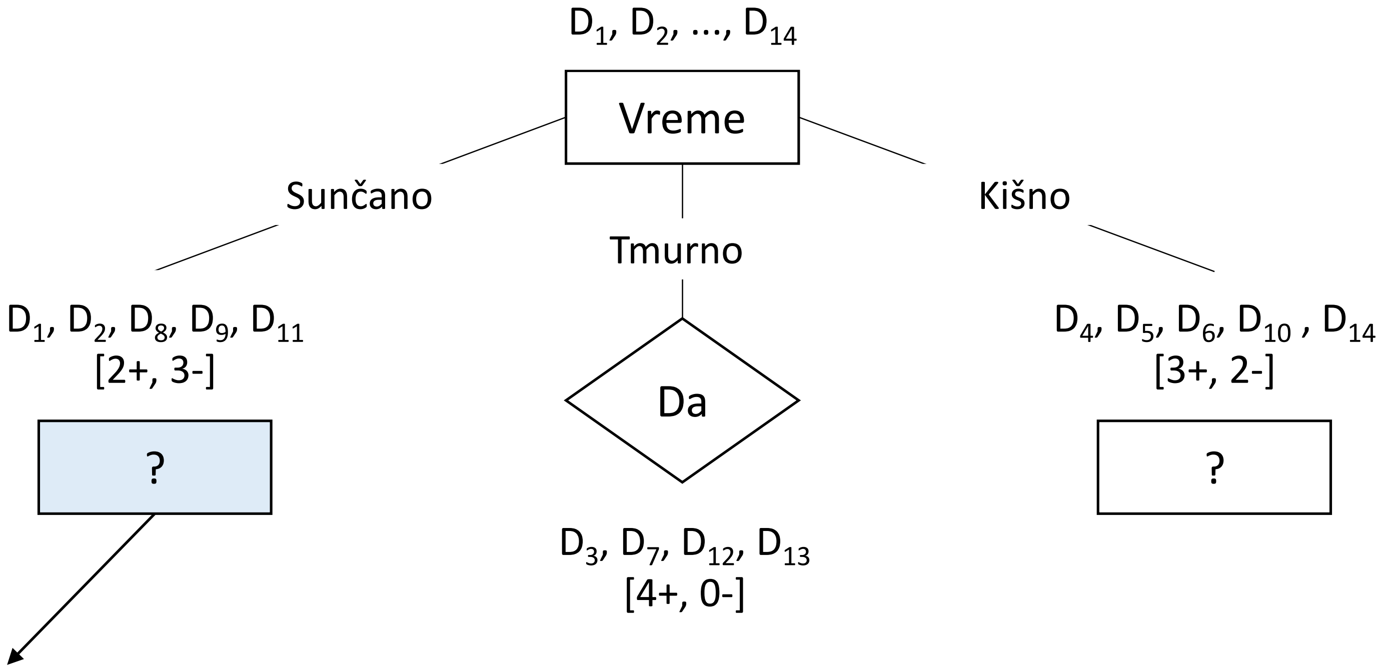
Traženje najboljeg atributa za klasifikaciju

Primer.

- Proces selektovanja novog atributa i podela klase obučavajućih primera se ponavlja za svaki sledeći neterminalni čvor, ovaj put koristeci samo obučavajuće primere koji se odnose na taj čvor.
- <u>Atributi koji su već iskorišćeni u stablu se isključuju</u>, tako da se svaki atribut može pojaviti samo jednom u stablu odluke duž bilo kog puta.
- Ovaj proces se nastavlja za svaki novi list dok se ne ispuni jedan od sledecih uslova:
  - (1) svaki atribut je vec iskorišćen duž ovog puta kroz stablo, ili
  - (2) svi obučavajući primeri koji se odnose na ovaj list imaju istu vrednost ciljnog atributa, tj. entropija je nula.
- Slika na str. 30 ilustruje izracunavanje informacionog dobitka za sledeći korak u rastućem stablu odluke.
- Krajnje stablo odluke koje je ID3 algoritam naučio iz 14 obučavajućih primera je dato na str. 7.

Traženje najboljeg atributa za klasifikaciju

Primer.



$$S_{sunčano} = \{D_1, D_2, D_8, D_9, D_{11}\}$$
$$Gain(S_{Sunčano}, Vlažnost) = 0{,}970$$
$$Gain(S_{Sunčano}, Temperatura) = 0{,}570$$
$$Gain(S_{Sunčano}, Temperatura) = 0{,}019$$

Učenje stabala odluke u prostoru hipoteza

Formulacija problema.

- Posmatrajmo proces nalaženja stabla odluke za zadate obučavajuce primere kao pretragu u prostoru svih hipoteza (stabala).
- Kako se može pronaći traženo stablo odluke?
- Jedan način bi bio da se pronađu sva moguća stabla za određeni skup atributa i njihove vrednosti, a zatim da se izabere najbolje stablo.
    - Ovaj postupak nije praktičan, jer broj mogućih stabala može biti jako veliki.
- Drugi mogući pristup bio bi: nađi sva moguća stabla koja klasifikuju zadati skup primera tačno, a zatim izaberi najjednostavnije stablo.
    - I ovaj pristup je velike računarske kompleksnosti.

Učenje stabala odluke u prostoru hipoteza

Podsetnik.

- Rekapitulirajmo sada kako ID3 algoritam za dve klase radi:
  - (1) Ako svi primeri pripadaju istoj klasi, kreiraj list.
  - (2) Inače, nađi najbolji atribut A, dodaj granu za svaku vrednost atributa A, rasporedi primere u podskupove.
  - (3) Ako su svi primeri klasifikovani tačno, stop.
  - (4) Inače, primeni korake 1-3 za listove.

Učenje stabala odluke u prostoru hipoteza

Pretraga prostora hipoteza.

- Kao i za druge induktivne metode, za ID3 se može reći da pretražuje prostor hipoteza da bi došao do one koja odgovara obučavajućim primerima.
- Prostor hipoteza koji ID3 pretražuje je skup mogućih stabala odluke.
- ID3 izvodi pretragu krećući se od jednostavnijih ka složenijim stablima kroz ovaj prostor hipoteza.
- Startuje se od praznog stabla, a nastavlja sa razmatranjem soženijih hipoteza u potrazi za stablom odluke koje tačno klasifikuje sve obučavajuće primere.
- Kriterijumska funkcija performanse koja usmerava ovu pretrage je informacioni dobitak.

Učenje stabala odluke u prostoru hipoteza

Uvid u sposobnosti i ograničenja ID3.

- Prostor hipoteza svih stabala odluke je potpun prostor konačnih funkcija diskretnih vrednosti.
  - Zato što svaku diskretnu funkciju može da predstavi stablom odluke, ID3 izbegava jedan od najvećih rizika kod metoda koje pretražuju nekompletne prostore hipoteza – mogućnost da prostor hipoteza ne sadrži ciljnu funkciju.

Učenje stabala odluke u prostoru hipoteza

Uvid u sposobnosti i ograničenja ID3.

- ID3 održava samo jednu trenutnu hipotezu u toku pretraživanja prostora stabala odluke.
  - Održavajući samo jednu hipotezu, ID3 gubi mogućnosti koje proizilaze iz eksplicitnog predstavljanja svih konzistentnih hipoteza.

Učenje stabala odluke u prostoru hipoteza

Uvid u sposobnosti i ograničenja ID3.

- ID3 u svojoj izvornoj formi ne može da se vrati na prethodne alternative u pretrazi.
    - Kada jednom selektuje atribut koji se testira na određenom nivou stabla, nikada se ne vraća da preispita taj izbor.
    - Stoga je podložan uobičajenim rizicima HC pretraga bez vraćanja – konvergiranje ka lokalno optimalnim rešenjima, koja nisu i globalno optimalna.

Učenje stabala odluke u prostoru hipoteza

Uvid u sposobnosti i ograničenja ID3.

- Da bi doneo odluke, ID3 koristi sve obučavajuće primere u svakom koraku pretrage.
  - Ovo je potpuno suprotno metodama koje odluke donose inkrementalno na bazi individualnih obučavajućih primera.
  - Prednost korišćenja statističkih osobina svih primera (npr. informacionog dobitka) je ta što su rezultati pretrage mnogo manje osetljivi na greške u individualnim obučavajućim primerima.
  - ID3 se lako može modifikovati režimu rada sa zašumljenim obučavajućim podacima, tako što se modifikuje njegov kriterijum završetka rada po kome se prihvataju i hipoteze koje ne moraju biti idealno konzistentne sa obučavajućim podacima.

Induktivni pomeraj

Kako je nešto uopšte moguće naučiti?

- Postavljamo suštinsko pitanje: kako je nešto uopšte moguće naučiti?
- Zašto bi neka procedura obučavanja odabrala konkretnu hiperpovršinu kao rezultat obučavanja na osnovu obučavajućeg skupa od zadate četiri tačke? Takvih površina ima beskonačno mnogo.
- Vrlo rano se u teorijskom razvoju mašinskog učenja shvatilo da je neophodno apriori uvesti neka ograničenja na prostor hipoteza u kome se traži finalno rešenje.
- Ova vrsta apriornih informacija se zove pomeraj (engl. *bias*).
- Obučavanje od nekog praktičnog značaja nije moguce bez pomeraja.

Induktivni pomeraj

Šta je induktivni pomeraj?

- Induktivni pomeraj je skup pretpostavki koje zajedno sa obučavajućim primerima opravdavaju klasifikacije budućih primera, koji nisu učestvovali u fazi obučavanja.
- Preciznije, induktivni pomeraj su restrikcije koje namećemo prostoru hipoteza, smatrajući zadati algoritam obučavanja uspešnim ukoliko je pronašao najbolju hipotezu u prostoru hipoteza uz istovremeno zadovoljavanje nametnutih restrikcija.

Induktivni pomeraj i stabla odlučivanja.

- Za skup obučavajućih primera postoji veliki broj stabala odluke koja su konzistentna sa ovim primerima.
- Opis induktivnog pomeraja algoritma ID3 se stoga sastoji u opisu pomeraja na osnovu koga on <u>bira jednu od konzistentnih hipoteza među svim ostalima</u>.
- ID3 bira prvo prihvatljivo stablo odluke na koje naiđe u svojoj pretrazi.
- Strategija pretraživanja metode ID3 (a) bira <u>kraća stabla</u> pre nego duža, i (b) bira stabla kod kojih su <u>atributi sa najvećim informacionim dobitkom postavljani najbliže korenu</u>.
- Zbog malih, ali važnih interakcija između heuristike za selekciju atributa koju ID3 koristi i konkretnih obučavajućih primera sa kojima se susreće, teško je precizno definisati njegov induktivni pomeraj.
- Pomeraj možemo približno definisati kao <u>davanje prednosti kratkim stablima odluke</u> nad kompleksnim stablima.

Induktivni pomeraj

Induktivni pomeraj i BFS-ID3.

- Zamislimo algoritam koji počinje prazim stablom i pretražuje po širini (engl. *breadth-first*) progresivno ka kompleksnijim stablima, razmatrajuci prvo sva stabla dubine 1, zatim 2 i tako redom.
- Kada pronađe stablo odluke konzistentno sa obučavajućim podacima, vraća najmanje stablo koje je konzistentno na toj dubini pretrage (npr. stablo sa najmanjim brojem čvorova).
- Nazovimo BFS-ID3 ovakav algoritam koji pretražuje po širini.
- BFS-ID3 pronalazi najkraće stablo odluke i time precizno primenjuje pomeraj “kraća stabla imaju prednost nad većima”.

Induktivni pomeraj

Induktivni pomeraj i ID3.

- ID3 se može posmatrati kao efikasna aproksimacija algoritma BFS-ID3, koja koristi pohlepnu heurističku pretragu u pokušaju da pronađe najkraće stablo bez sprovođenja čitave pretrage po širini kroz prostor hipoteza.
- Pošto ID3 koristi heuristiku informacionog dobitka i HC strategiju pretrage, on pokazuje složeniji pomeraj nego BFS-ID3.
- Konkretno, on ne pronalazi najkraće stablo odluke svaki put, i naklonjen je stablima kod kojih su atributi sa najvecim informacionim dobitkom postavljeni što bliže korenu.

Induktivni pomeraj

Pomeraj prednosti i restriktivni pomeraj.

- Induktivni pomeraj ID3 algoritma je prednost koja se daje nekim hipotezama u odnosu na druge, bez striktnih ograničenja na hipoteze koje kasnije mogu biti generisane.
  - Ova vrsta pomeraja se naziva pomeraj prednosti ili, alternativno, pomeraj pretrage.
- Nasuprot tome, postoje algoritmi čiji je pomeraj u vidu kategoričke restrikcije na skup razmatranih hipoteza.
  - Takva vrsta pomeraja se naziva restriktivni pomeraj.
- Obično je pomeraj prednosti poželjniji od restriktivnog pomeraja, jer dozvoljava obučavajućem sistemu da radi u celokupnom prostoru hipoteza, koji sigurno sadrži ciljnu funkciju.
- Restriktivni pomeraj, sa druge strane, striktno ograničava skup potencijalnih hipoteza, pa se javlja mogućnost da nepoznata ciljna funkcija nije u domašaju obučavajućeg sistema.

Induktivni pomeraj

Okamov rezač.

- Da li je činjenica da induktivno pomeraj algoritma ID3 daje prednost kraćim stablima odluke, razumna osnova za generalizaciju van obučavajućih primera?
- Filozofi i naučnici koji se bave tom oblašću su dugo vremena diskutovali o ovom pitanju, ali je za sada debata ostala nerešena.
- Vilijem Okam je jedan od prvih koji su razmatrali ovo pitanje oko 1320. godine, pa se ovaj pomeraj često naziva i Okamov rezač.
- Okamov rezač znači da za objašnjenje nekog fenomena treba uvesti što je moguće manje pretpostavki, eliminišući, tj. odsecajući kao rezačem, one pretpostavke koje ne doprinose predviđanjima hipoteze ili teorije.
- Kada više različitih teorija ima jednaku mogucnost predviđanja, princip preporučuje da se uvede što je moguće manje pretpostavki.

Praktični problemi učenja stabala odluke

Praktični problemi koji se javljaju kod stabala odluke.

- Praktični problemi koji se javljaju kod stabala odluke su:
  - određivanje koliko duboko treba razviti stablo odluke,
  - obrađivanje kontinualnih atributa,
  - odabir odgovarajuće mere za selekciju atributa,
  - rad sa obučavajućim podacima kojima nedostaju vrednosti atributa,
  - rad sa atributima koji imaju promenljive troškove,
  - unapređenje efikasnosti računanja.
- Posebno ćemo razmatrati neka od ovih pitanja kao i modifikacije osnovnog ID3 algoritma kojim se ovi problemi rešavaju ili ublažuju.
- Tako dobijen rezultujući sistem preimenovan je u algoritam poznat pod nazivom C4.5.

Praktični problemi učenja stabala odluke

Izbegavanje prenaučenosti (preprilagođenosti).

- Algoritam ID3 pušta svaku granu stabla da raste upravo toliko duboko da idealno klasifikuje obučavajuće primere.
- Iako je ovo ponekad razumna strategija, ona u stvari može da dovede do poteškoća kada su podaci zašumljeni, ili kada je broj obučavajućih primera suviše mali da bi predstvaljao reprezentativni uzorak stvarne ciljne funkcije.
- U bilo kom od ova dva slučaja, jednostavni algoritam može stvoriti stablo koje ima osobinu <u>preprilagođenosti</u> (engl. *overfitting*) datom obučavajućem skupu.
- Kaže se da je neka hipoteza preprilagodjena obučavajućem skupu, ako <u>postoji neka druga hipoteza</u>, koja je manje tačna na obučavajućem skupu, ali <u>daju veću tačnost na primerima van obučavajućeg skupa</u>.

Praktični problemi učenja stabala odluke

Izbegavanje prenaučenosti (preprilagođenosti).

- Preprilagodjenost je značajan praktican problem za učenje stabala odluke, ali i gotovo sve algoritme mašinskog učenja.
- Postoji više pristupa rešavanju problema preprilagođenosti kod učenja stabala odluke, a ovde ćemo navesti samo dva:
  - pristupi ranog zaustavljanja rasta stabla, pre nego što dostigne tačku gde idealno klasifikuje obućavajuce podatke;
  - pristupi koji dozvoljavaju da se desi preprilagođenje, a onda se naknadno vrši tzv. kresanje preprilagođenog stabla.
- Iako prvi pristup može delovati direktniji, drugi pristup je uspešniji u praksi, usled nemogućnosti tačne procene konačne veličine stabla kod ranog zaustavljanja.

Izbegavanje prenaučenosti (preprilagođenosti).

- Nezavisno da li se adekvatna veličina stabla postiže na osnovu prvog ili drugog pristupa, ključno pitanje je koji kriterijum treba koristiti za njeno određivanje.
- Pristupi su sledeći:
    - (1) Koristiti odvojen skup primera, različit od obučavajućeg, za procenu koristi od naknadnog kresanja čvorova sa stabla.
    - (2) Koristiti sve dostupne podatke za obuku, ali primeniti statistički test za procenu da li proširivanje ili kresanje određenog čvora dovodi do poboljšanja za celokupnu distribuciju primera ili samo za obučavajuće podatke.
    - (3) Koristiti eksplicitne mere kompleksnosti za kodovanje obučavajućih primera i stabla odluke, zaustavljajući rast stabla kada je kompleksnost kodovanja minimizirana.

Praktični problemi učenja stabala odluke

Kresanje sa umanjenom greškom.

- Pristup koji se naziva kresanje sa umanjenom greškom razmatra svaki čvor odluke u stablu kao kandidata za odbacivanje.
- Kresanje čvora odluke se sastoji od uklanjanja podstabla čiji je koren taj čvor, pravljenja lista i dodeljivanja tom listu oznake većinske klasifikacije obučavajućih primera koji su bili povezani sa čvorom koji krešemo.
    - Čvor se uklanja samo ako se rezultujuće okresano stablo ne pokaže lošije na validacionom skupu od polaznog stabla.
    - Čvorovi se krešu iterativno, uvek uzimajuci onaj čvor čijim se uklanjanjem najviše povećava tačnost stabla odluke na validacionom skupu.
    - Kresanje čvorova se nastavlja dok ne postane štetno, tj. dok ne počne da se smanjuje tačnost stabla na validacionom skupu.

Uvođenje atributa sa kontinualnim vrednostima.

- Početna definicija ID3 algoritma je ograničena na diskretne atribute koji uzimaju vrednosti iz konačnih skupova.
  - Prvo, ciljni atribut čija vrednost se predviđa stablom odluke mora biti diskretan.
  - Drugo, atributi koji se testiraju u čvorovima stabla odluke moraju takođe biti diskretni.
- Ovo drugo ogranicenje se lako može ukloniti, tako da stabla odluke podržavaju rad sa kontinualnim atributima.
- To se rešava pomocu dinamičke diskretizacije atributa, tj. podele opsega kontinualnog atributa u disjunktne intervale kojima se zatim dodeljuju diskretne vrednosti.
- Npr. za atribut A koji je kontinualan, algoritam može dinamički da formira novi Bulov atribut A_c, koji je tačan ako je $A < c$ i netačan u ostalim slucajevima.
- Pitanje je kako odrediti vrednost praga c?

Uvođenje atributa sa kontinualnim vrednostima.

- Kao primer, pretpostavimo da želimo da uključimo novi kontinualni atribut *Temperatura* u vektore obeležja obučavajućeg skupa.
- Pretpostavimo da obučavajući primeri u vezi sa nekim određenim čvorom u stablu odluke imaju sledeće vrednosti za *Temperatura* i ciljni atribut *IgratiTenis*:

Temperatura	4,5	9	15,5	22	26,5	32
IgratiTenis	Ne	Ne	Da	Da	Da	Ne

Praktični problemi učenja stabala odluke

Uvođenje atributa sa kontinualnim vrednostima.

- Poželjno bi bilo da odaberemo prag c takav da se dobije najveci informacioni dobitak.
- Sortiranjem vrednosti kontinualnog atributa A, a potom identifikovanjem mesta promene ciljne klasifikacije možemo generisati skup vrednosti koje su kandidati za prag.
- Pragove računamo kao sredinu intervala između odgovarajućih vrednosti atributa A na mestima promene ciljne klasifikacije:
  - (9+15,5)/2=12,5 i
  - (26,5+32)/2=29,5.
- Potom računamo informacioni dobitak za svaki od atributa kandidata: *Temperatura*>12,25 i *Temperatura*>29,25, a zatim biramo veći, koji je u ovom primeru *Temperatura*>12.25.
- Ovakav dinamički kreiran Bulov atribut se priključuje drugim diskretnim atributima u procesu formiranja stabla odluke.
- Izloženi pristup se može generalizovati sa binarne na opštu, n-arnu diskretizaciju.

Praktični problemi učenja stabala odluke

Alternativne mere za selektovanje atributa.

- Jedan od nedostataka mere informacionog dobitka je <u>favorizovanje atributa sa više vrednosti</u> nad onima sa manjim brojem vrednosti.
- Kao ekstreman primer, razmotrimo atribut *Datum*, koji ima veliki broj mogućih vrednosti.
- Kada bismo ovaj atribut dodali u našu tabelu, on bi imao <u>najveći informacioni dobitak</u> među svim ostalim atributima.
- To sledi iz činjenice da *Datum* savršeno predviđa ciljni atribut za obučavajuće podatke.
  - Stoga bi *Datum* bio odabran za atribut odluke za koreni čvor stabla i na taj način bismo dobili <u>veoma široko stablo dubine 1</u>, koje idealno klasifikuje obučavajuće primere.
  - Ovakvo stablo odluke bi se loše pokazalo na sledujućim primerima, jer <u>nije koristan prediktor</u> uprkos tome što odlično razdvaja obučavajuće podatke.

Alternativne mere za selektovanje atributa.

- Jedan način da se ova prepreka prebrodi je da se selektuju atributi bazirani na meri koja nije informacioni dobitak.
- Alternativna mera koja se uspešno koristi je stepen dobitka (engl. *gain ratio*).
- Stepen dobitka kažnjava atribute kao što je *Datum* tako što uključuje pojam podeljene informacije (engl. *split information*), koji je osetljiv na to koliko uniformno i široko atribut deli podatke:

$$SplitInformation(S, A) = -\sum_{i=1}^{c} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

- gde su S_1 do S_c podskupovi primera koji nastaju particijom skupa S atributom A, čiji je broj vrednosti jednak c.

Alternativne mere za selektovanje atributa.

- Mera *GainRatio* se može izraziti preko ranije definisanih mera *Gain* i *SplitInformation*:

$$GainRatio(S, A) = \frac{Gain(S, A)}{SplitInformation(S, A)}$$

- Primetimo da *SplitInformation* <u>zabranjuje selekciju atributa sa verlikim brojem uniformno raspoređenih vrednosti</u>.
  - Pogledajmo, na primer, kolekciju od *n* primera koji su potpuno razdvojeni atributom *A* (npr. *Datum*) – u ovom slučaju, vredenost *SplitInformation* bi bila $\log_2 n$.
  - Nasuprot tome, Bulov atribut *B* koji razdvaja istih *n* primera tačno na pola bi imao *SplitInformation*=1.
- Ako atributi A i B daju isti informacioni dobitak, onda je očigledno da će B imati veći stepen dobitka.

Alternativne mere za selektovanje atributa.

- Jedan praktičan problem koji dovodi u pitanje korišćenje mere stepen dobitka je to što imenilac u izrazu može biti za neko S_i jednak nuli ili veoma mali kada je $|S_i| \approx |S|$.
- Zbog toga vrednost *GainRatio* može biti ili nedefinisana ili veoma velika kod atributa za koje se dogodi da imaju skoro iste vrednosti za sve članove skupa S.
- Da bismo izbegli biranje atributa samo po ovoj osnovi, možemo usvojiti neke heuristike, kao što je prvo racunati *Gain* svakog atributa ponaosob, a onda primeniti *GainRatio* test, razmatrajući samo one atribute sa vrednošcu *Gain* iznad prosečne vrednosti.

Praktični problemi učenja stabala odluke

Obučavajući primeri kod kojih nedostaju vrednosti atributa.

- U nekim slučajevima kod obučavajućih podataka mogu da nedostaju vrednosti nekih atributa.
- Na primer, u domenu medicine kada želimo da predvidimo ishod za nekog pacijenta na bazi raznih iscrpnih testova, može se desiti da je test *RezultatAnalizeKrvi* dostupan samo za podskup pacijenata.
- U takvim slučajevima, uobičajeno je da se atributi koji nedostaju estimiraju na osnovu ostalih primera čiji atributi imaju poznate vrednosti.

Obučavajući primeri kod kojih nedostaju vrednosti atributa.

- Posmatrajmo situaciju u kojoj treba izracunati $Gain(S, A)$ u *n*-tom čvoru stabla odluke, da bi se procenilo da li je atribut A najbolji atribut za testiranje u ovom cvoru.
- Pretpostavimo da je $\langle x, c(x) \rangle$ jedan od obučavajućih primera u skupu S i da vrednost $A(x)$ nije poznata.
- Jedna strategija rešavanja problema je da mu se dodeli vrednost koja se najcešce javlja među obučavajućim primerima u *n*-tom čvoru.
- Alternativno, možemo mu dati vrednost koja se najčešce pojavljuje među primerima u *n*-tom čvoru i koja istovremenoima klasifikaciju $c(x)$.
- Potom, algoritam učenja stabla odluke direktno može da koristi ovakav obučavajući primer koji je dopunjen procenjenom vrednošcu $A(x)$.

Obučavajući primeri kod kojih nedostaju vrednosti atributa.

- Druga, kompleksnija procedura, koju koristi C4.5, je da se dodeli verovatnoća svakoj od mogućih vrednosti atributa A, a ne jednostavno biranje najcešce vrednosti $A(x)$.
- Ove verovatnoće se procenjuju na osnovu ucestanosti pojavljivanja svake vrednosti od A za n-ti cvor.

Obučavajući primeri kod kojih nedostaju vrednosti atributa.

- Na primer, za Bulov atribut A, ako *n*-ti čvor ima 6 poznatih vrednosti za $A = 1$ i 4 za $A = 0$, rekli bismo da su verovatnoce $P(A(x) = 1) = 0.6$ i $P(A(x) = 0) = 0.4$.
- Deo od 0,6 primera x se raspoređuje duž grane za $A = 1$, a preostalih 0,4 niz drugu granu.
- Ovi frakcioni primeri se koriste radi računanja informacionog dobitka i mogu se dalje deliti na nove grane ako dođe do testiranja još nekog atributa čije vrednosti nedostaju.
- Isto frakcionisanje primera može biti primenjeno i nakon učenja, da bi se klasifikovali novi primeri nepoznatih vrednosti atributa.
- U ovom slucaju, klasifikacija novih primera je najverovatnija klasifikacija, a računa se sumiranjem težina fragmenata klasifikovanih na različite načine u listovima stabla.

Atributi sa promenljivom cenom.

- U nekim zadacima učenja primeri se mogu vezivati i za cenu (engl. *cost*).
- Prilikom učenja automatske dijagnostike u medicinskom domenu, pacijente možemo opisati atributima kao što su *Temperatura*, *RezultatiBiopsije*, *Puls*, *RezultatAnalizeKrvi* i slično.
- Cene ovih atributa značajno variraju, kako u pogledu novčanih troškova takao i uticaja na zdravlje.
- Kod takvih zadataka, više odgovaraju stabla odluke koja koriste atribute sa niskom cenom gde god je moguće, a da se oslanjaju na atribute sa visokom cenom samo gde je to neophodno da bi se došlo do pouzdane klasifikacije.
- ID3 se može modifikovati da uzima u obzir cene atributa tako što se uvodi cena u meru selekcije atributa.
- Na primer, možemo podeliti informacioni dobitak cenom atributa, tako da se više koriste atributi sa nižom cenom.

Literatura

- Beleške pripremljene prema knjizi – Milan Milosavljević (2015): „Veštačka inteligencija“. Univerzitet Singidunum, Beograd.

Hvala na pažnji

Pitanja su dobrodošla.