1 Osnove

1.1 Kaj je umetna inteligenca?

- **cilji:** Razumeti in zgraditi inteligentne sisteme na osnovi razumevanja cloveskega *razmisljanja, sklepanja, ucenja* in *komuniciranja.*

2 Strojno Ucenje

2.1 Kaj je stronjo ucenje?

Je *podrocje umetne inteligence*, ki raziskuje kako se lahko algorimti samodejno izboljusjejo ob pridobivanju izkusenj.

2.2 Vrste ucenja:

- **Nadzorovano ucenje** *supervised learning*: Ucni primeri so oznaceni in podani kot vrednosti vhodov in izhodov. Ucimo se funkcije, ki vhode preslika v izhode. (npr. odlocitveno drevo)
- **Nenadzorovano ucenje** *unsupervised learning*: Ucni primeri niso oznaceni → nimajo ciljne spremenljivke. Ucimo se iz vzorcev v podatkih. (npr. grucenje)
- **Spodbujevalno ucenje** *reinforcment learning*: Inteligentni agen se uci iz zaporedja nagrad in kazni.

2.3 Nadzorovano ucenje

Podano imamo mnozico **ucnih** primerov:

$$(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$$

kjer je vsak y_j vrednost neznane funkcije $y = f(x)$. Nasa naloga je posikati hipoteticno funkcijo h, ki je najboljsi mozen priblizek funkciji f.

Locimo dve vrsti problemov:

1. **Klasifikacijski**: y_j je *diskretna(kategoricna)* spremenljivka
   - y pripada **koncnemu naboru vrednosti** (diskretna spremenljivka)
   - y imenujemo **razred** (class)
2. **Regresijski**: y_j je *zvezna* spremenljivka
   - y je stevilo (obicajno $y \in R$, je zvezna spremenljivka)
   - y imenujemo **oznacba** (label)

2.3.1 Prostor in evalviranje hipotez

Denimo da imamo:

- binarno klasifikacijo
- n binarnih atributov

Iz tega sledi:

- 2^n razlicnih ucnih primerov
- 2^{2^n} hipotez (celotno odlocitveno drevo)

Pomembni kriteriji pri *evalviranju* hipotez:

- **konsistentnost** hipotez s (ucnimi) primeri
- **splosnost** tocnost za nevidene primere
- **razumljivost** (*interpretability, comprehensibility*) hipotez

Poznamo 4 razrede za ocenjevanje uspesnosti pri klasifikaciji na podlagi njihove **tocnosti**:

- **TP** - pravilno pozitivno klasificirani primeri
- **TN** - pravilno negativno klasificirani primeri
- **FP** - napacno pozitivno klasificirani primeri
- **FN** - napacno negiativno klasificirani primeri

Klasifikacijska tocnost je potem definirana:

$$CA = \frac{TP+TN}{TP+TN+FP+FN} = \frac{TP+TN}{N}$$

Poznamo dva tipa atributov:

1. **diskretni** atributi:
   - **nominalni** npr. ['soncno', 'dezevno']
   - **ordinalni** npr. ['nizko', 'srednje', 'visoko']

   Odlocitvena drevesa delijo prvotno ucno mnozico na vse manjse podmnozice
2. **zvezni** atributi:

   Delitev podmnozice glede na smislno mejo izbranega atributa

2.3.2 Odlocitveno drevo

Ponazarja relacijo med vhodnimi *vrednostmi/atributi* in *odlocitvojo/ciljno* spremenljivko.

Z **notranjimi vozlisci** opravljamo test glede na vrednost posameznega atributa. Na koncu pridemo do **lista**, ki nam s poroci odlocitev (vrednost ciljne spremenljivke). Konjunkcijo pogojev v *notranjih vozliscih* katera vodi do *lista* imenujemo **pot**.

Gradnja odlocitvenega drevesa: Nas cilj je zgraditi **cim manjse drevo**, ki je **konsistentno** z ucnimi podatki.

Hevristicni pozrezsni algoritem - TDIDT s strategijo **razveji in omeji**:

- Izberi najbolj pomemben atribut - tisti, ki najbolj odlocilno vpliva na klasifikacijo primera in razdeli primere v poddrevesa glede na njegove vrednosti
- rekurizvno ponovi za vsa drevesa
- ce vsi elementi v listu pripadajo istemu razredu ali vozlisca ni mozno deliti naprej(ni razpolozljivih atributov), ustavi gradnjo

Kratovidnost TDIDT: Ker je TDIDT pozresni algoritem, ki "lokalno" izbira najboljsi atribut, ne upsteva kako dobro drugi algoritmi doplnjujejo izbrani atribut.

2.3.2.1 Izbor najbolj pomembnega atributa in informacijski prispevek

Najboljsi atribut je tisti, ki razdeli ucno mnozico v najbolj ciste podmnozice. Uporabimo lahko mero entropije:

$$H(X) = \sum_{i=1}^{n} p_i I_i = -\sum_{i=1}^{n} p_i log_2 p_i$$

Zanima nas **znizanje** entropije (*nedolocenosti*) ob delitvi ucne mnozice glede na vrednosti atributa A.

Definirajmo **informacijski prispevek** na taksen nacin, da najbolj informativni atribut **maksimizira informacijski prispevek** oz. minimizira I_{res}.

$$Gain(A) = I - I_{res}(A)$$
$$I_{res} = -\sum_{v_i \in A} p_{v_i} \sum_c p(c|v_i) log_2 p(c|v_i)$$

2.3.2.2 Vecvrednostni atributi

Tezava z atributi, ki imajo vec kot dve vrednosti: Informacijski prispevek precenjuje njihovo kakovost(entropija je visja na racun vecjega stevila vrednosti in ne na racun kakovosti atributa)

resitve:

- normalizacija informacijskega prispevka: **relativni informacijski prispevek** ali IGR (information gain ratio)

  $$Gain(a) = I - I_{res}(A),\ I(A) = -\sum_v p_v log_2 p_v$$
  $$GainRatio(A) = \frac{Gain(A)}{I(A)} = \frac{I - I_{res}(A)}{I(A)} \text{ Oba zelimo maksimizirati}$$

- uporaba **alternativnih mer**: npr. **Gini index** *Ocena pricakovane klasifikacijske napake, vsota produktov verjetnosti razredov*

  $$Gini = \sum_{c1 \neq c2} p(c_1)p(c_2)$$
  $$Gini(A) = \sum_v p(v) \sum_{c1 \neq c2} p(c_1|v)p(c_2|v)$$

- **binarizacija** atributov: Je alternativa za resevanje problematike z vecvrednostmi atributi. Prednosti binarizacije so manjsa vejanja drevesa, kar je statisticno bolj zanesljivo. Razlicni nacini binarizacije atributa lahko nastopajo kot samostojni atributi, ki se v drevesu pojavijo veckrat.

2.3.2.3 Privzeta tocnost in Pristranost na ucni mnozici

Smiselna mera za **Privzeto tocnost** odlocitvenega drevesa je **verjetnost vecinskega razreda** v ucni mnozici. Drevo je uporabno, ce je njegova tocnost **visja** od privzete tocnosti.

npr. [#Da, #Ne] = [3, 7] $\rightarrow$ verjetnost vecinskega razreda: 7/10

Nas cilj je maksimizirati pricakovano tocnost na testnih podatkih vendar se zelimo izogniti pretiranemu prilaganju. Zato obicajno podatke razdelimo na **ucno** (70%) in **testno** mnozico (30%).

2.3.2.4 Ucenje dreves iz sumnih podatkov

V primeru da podatki niso popolni(premalo primerov / atributov) ali napake se lahko pojavijo tezave:

- **Ucenje suma** in ne dejanske aproksimacijske funkcije
- Pretirano prilaganje vodi v **prevelika drevesa** $\rightarrow$ overfitting
- **Slaba razumljivost** dreves

Posledica: **nizja klasifikacijska tocnost** na novih nevidenih podatkih.

2.3.2.5 Rezanje odlocitvenih dreves

Resujemo problem prevelikega prilaganja ucnim podatkom. Nizji deli drevesa predstavljajo vecje lokalno prilaganje ucnim podatkom, ki so lahko posledica suma. Poznamo dve strategiji rezanja dreves:

1. **Rezanje vnaprej**: uporaba dodatnega kriterija za zaustavitev gradnje devesa glede na obseg suma. Je hitrejsi pristop, vendar kratkoviden (uposteva samo zgornji del drevesa)
2. **Rezanje nazaj**: Rezanje, ki po gradnji celotnega drevesa, odstrani manj zanesljive dele drevesa(opisujejo sum, zgrajeni iz manj podatkov in z manj informirani atributi) Je pocasneje, in upostecva informacijo iz **celega drevesa**.

1. **Rezanje z zmanjsevanjem napake (REP)**

   Uporablja posebno reazalno (validacijsko) mnozico:

   - **Ucna mnozica** (70%):
     - mnozica za gradnjo *growing set* (70%)
     - rezalna mnozica *prunning set* (30%)
   - **Testna mnozica** (30%)

   Postopek:

   (a) Potuj od listov navzgor(pricni s starsi listov)

   (b) Za vsako vozlisce **v** izracunaj **dobitek rezanja**

   Dobitek rezanja izracunamo: st. napacnih klasifikacij v drevesu **T** - st. napacnih klasifikacij v vozliscu **v**.

   (c) Ce je dobitek ≥ 0, obrezi in nadaljuj postopek s starsem, sicer ustavi postopek

2. **Rezanje z minimizacijo napake (MEP)**

   Uporablja mnozico za gradnjo drevesa in ne locene rezalne mnozice. Postopek: Za vozlisce **v** izracunamo:

   (a) **staticno napako** (verjetnost klasifikacije v napacen razred)

   $$e(v) = p(razred \neq C|v), \text{ C je vecinski razred v } \mathbf{v}$$

   (b) **vzvratno napako** (*backed-up error*)

   $$\sum_i p_i E(T_i) = p_1 E(T_1) + p_2 E(T_2) + \ldots$$

   Rezemo, ce je **staticna napaka** < **vzvratna napaka**.

   Napaka **optimalno** obrezanega drevesa je torej:

   $$E(T) = min(e(v), \sum_i p_i E(T_i)) \text{ in } E(T) = e(v), \text{ ce je } \mathbf{v} \text{ list.}$$

   Namesto **minimizacije** napake E, lahko provblem obrnemo in **maksimiziramo** tocnost CA.

2.3.2.6 Ocenjevanje verjetnosti

1. **Laplaceova ocena verjetnosti**

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

   n - st. primerov, ki pripadajo razredu C
   N - st. vseh primerov
   k - st. vseh razredov
   k je problematicen parameter, saj ocena ne uposetva apriorne verjetnosti

2. **m-ocena verjetnosti**

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

   p_a - Apriorna verjetnost razreda C
   m - parameter ocene (vpliva na delez upostevanja apriorne verjetnosti)
   Ce imamo malo suma, potem m nastavimo na majno vrednost in imamo malo rezanja. Obratno v primeru ce imamo veliko suma. Gre se za posplositev Laplaceove ocene za $m = k$ in $p_a = 1/k$.