Uvod u umjetnu inteligenciju – 2020/2021 ZI

Grupa F

1 (T) Bayesovo pravilo omogućava nam da na temelju opaženog dokaza E zaključimo o vjerojatnosti hipoteze H. Iz perspektive logike, takvo zaključivanje odgovara pravilu abdukcije. **Kojem dijelu pravila abdukcije odgovara uvjetna vjerojatnost $P(E|H)$ iz Bayesovog pravila?**

A Implikaciji $H \to E$ B Implikaciji $E \to H$ C Činjenici E D Činjenici H

2 (T) Neizrazita logika temelji se na teoriji neizrazitih skupova. **Koja je točno veza između neizrazite logike i neizrazitih skupova?**

A Atom $P(x)$ istinit je za one i samo one elemente za koje $\mu(x) \geq 0.5$

B Disjunkcija neizrazitih skupova P i Q jednaka je neizrazitom skupu sa $\mu(x) = \max(\mu_P(x), \mu_Q(x))$

C Vrijednost $\mu(x)$ je donja ograda vjerojatnosti da je atom $P(X)$ istinit

D Stupanj istinitosti atoma $P(x)$ jednak je $\mu_P(x)$, tj. stupnju pripadnosti elementa x neizrazitom skupu P

3 (P) Za zaključivanje u domeni pedijatrije koristimo Bayesovo pravilo. Znamo da je vjerojatnost nastupanja osipa ako dijete ima šarlah deset puta veća nego vjerojatnost nastupanja osipa ako dijete nema šarlah. Također znamo da je vjerojatnost šarlaha ako dijete ima osip barem dva puta veća nego vjerojatnost šarlaha prije opažanja osipa. **Kolika je vjerojatnost P šarlaha prije opažanja osipa?**

A $1/2 \leq P \leq 2/3$ B $P \leq 4/9$ C $P \geq 1/6$ D $P = 4/5$

$P(O|\check{S}) = 10P(O|\neg\check{S})$

$P(\check{S}|O) \geq 2P(\check{S})$

$$\frac{P(O|\check{S})P(\check{S})}{P(O)} \geq 2P(\check{S})$$

$$\frac{P(O|\check{S})P(\check{S})}{P(O|\check{S})P(\check{S}) + P(O|\neg\check{S})P(\neg\check{S})} \geq 2P(\check{S})$$

$$\frac{10\,P(O|\neg\check{S})P(\check{S})}{10\,P(O|\neg\check{S})P(\check{S}) + P(O|\neg\check{S})(1 - P(\check{S}))} \geq 2P(\check{S})$$

$$\frac{10\,P(\check{S})}{10\,P(\check{S}) + 1 - P(\check{S})} \geq 2P(\check{S})$$

$$\frac{10\,P(\check{S})}{9\,P(\check{S}) + 1} * \frac{9\,P(\check{S}) + 1}{P(\check{S})} \geq 2P(\check{S}) * \frac{9\,P(\check{S}) + 1}{P(\check{S})}$$

$$10 \geq 18\,P(\check{S}) + 2$$

$$P(\check{S}) \leq \frac{4}{9}$$

4 (P) Primjenom standardnih (Zadehovih) neizrazitih operacija i jezičnih modifikatora konstruirali smo neizraziti skup *vrlo jak klokan*. Klokan Roger, najjači australski klokan, koji je od starosti preminuo 2018. godine, tom skupu pripada sa $\mu = 0.9$. Primjenom istih operatora i modifikatora konstruirali smo neizraziti skup *ne jak klokan*. **Koja je pripadnost klokana Rogera tom neizrazitom skupu?**

A $\sqrt{1-0.9^2}$ B $1-\sqrt{0.9}$ C $1-0.9^2$ D $(1-0.9)^2$

$$\mu_{vrlo jak} = 0.9$$

Koncentracija Con(A) (jezični izraz VRLO) je:

$$\mu_{vrlo\ jak} = {\mu_{jak}}^2$$

Stoga da bi dobili pripadnost skupu jak (bez VRLO), korjenujemo:

$$\mu_{jak} = \sqrt{\mu_{vrlo\ jak}}$$

Pripadnost skupu NE jak je:

$$\mu_{ne\ jak} = 1 - \mu_{jak}$$

Stoga je

$$\mu_{ne\ jak} = 1 - \sqrt{0.9}$$

5 (R) Razvijamo sustav neizrazitog zaključivanja za modeliranje dinamike fluida. Razmatramo odnos tlaka, temperature i volumena fluida. Definirali smo univerzalne skupove $P = \{100, 200, 300, 400, 500\}$ za tlak (u Pascalima), $T = \{-100, -50, 0, 50, 100\}$ za temperaturu (u Kelvinima) i $V = \{0, 5, 10, 15, 20, 25\}$ za volumen (kubni metar po molu). Nad tim smo skupovima definirali neizrazite skupove *visok tlak* kao $P_v = \{0.1/200, 0.3/300, 0.6/400, 1/500\}$, skup *visoka temperatura* kao $T_v = \{0.2/0, 0.5/50, 0.8/100\}$, i skup *malen volumen* kao $V_m = \{1/5, 0.5/10, 0.3/15\}$. Definirali smo i dva pravila (implikacije), $P_v \to V_m$ i $T_v \to P_v$, modelirana kao neizrazite relacije. Zanima nas koliko je malen volumen fluida ako je temperatura fluida vrlo visoka, tj. želimo izvesti neizraziti skup V'_m ako je premisa neizraziti skup $T'_v = vrlo(T_v)$. To možemo izračunati primjenom Zadehovog modifikatora intenzifikacije te uzastopnom primjenom generaliziranog modusa ponensa. **Kako glasi neizraziti skup V'_m koji dobivamo takvim neizrazitim zaključivanjem?**

A $\{0.6/5, 0.3/10, 0.3/15\}$ C $\{0.6/5, 0.3/10, 0.1/15\}$

B $\{0.64/5, 0.5/10, 0.3/15\}$ D $\{0.64/5, 0.5/10, 0.1/15\}$

P	200	300	400	500
P_v	0.1	0.3	0.6	1

T	0	50	100
T_v	0.2	0.5	0.8

V	5	10	15
V_m	1	0.5	0.3

$$T_v \rightarrow P_v \rightarrow V_m$$

Svaka implikacija predstavlja neku neizrazitu relaciju.
Implikacija *"Ako x je **A** onda y je **B**"* određuje **neizrazitu relaciju** $\mu_{\mathbf{A \times B}}(x,y) = \min(\mu_{\mathbf{R}}(x), \mu_{\mathbf{R}}(y))$

$$R_1 = \mu_{\boldsymbol{T_v} \times \boldsymbol{P_v}}$$

	P	200	300	400	500
T	$T_v \setminus P_v$	0.1	0.3	0.6	1
0	0.2	0.1	0.2	0.2	0.2
50	0.5	0.1	0.3	0.5	0.5
100	0.8	0.1	0.3	0.6	0.8

npr. $\mu_{\boldsymbol{T_v} \times \boldsymbol{P_v}}(50{,}300) = min\{0.5{,}0.3\} = 0.3$

$$\boldsymbol{R_2} = \mu_{\boldsymbol{P_v} \times \boldsymbol{V_m}}$$

	V	5	10	15
P	$P_v \setminus V_m$	1	0.5	0.3
200	0.1	0.1	0.1	0.1
300	0.3	0.3	0.3	0.3
400	0.6	0.6	0.5	0.3
500	1	1	0.5	0.3

$T'_v = vrlo(T_v) = T_v^2$

T_v	0	50	100
T'_v	0.04	0.25	0.64

$P'_v = T'_v \circ R_1$

Za svaki element skupa x u $P'_v(x)$, radimo skalarni umnožak vektora T'_v i pripadnog stupca x u R_1, osim što umnožak zamijenimo izrazom min(), a zbroj izrazom max():

$\mu_{P'_v}(200) = max\{\, min(0.04, 0.1), min(0.25, 0.1), min(0.64, 0.1)\} = max\{0.04, 0.1, 0.1\} = 0.1$
$\mu_{P'_v}(300) = max\{\, min(0.04, 0.2), min(0.25, 0.3), min(0.64, 0.3)\} = max\{0.04, 0.25, 0.3\} = 0.3$
$\mu_{P'_v}(400) = max\{\, min(0.04, 0.2), min(0.25, 0.5), min(0.64, 0.6)\} = max\{0.04, 0.25, 0.6\} = 0.6$
$\mu_{P'_v}(500) = max\{\, min(0.04, 0.2), min(0.25, 0.5), min(0.64, 0.8)\} = max\{0.04, 0.25, 0.64\} = 0.64$

P	200	300	400	500
P'_v	0.1	0.3	0.6	0.64

$V'_m = P'_v \circ R_2$

$\mu_{V'_m}(5) = max\{min(0.1, 0.1), min(0.3, 0.3), min(0.6, 0.6), min(0.64, 1)\} = max\{0.1, 0.3, 0.6, 0.64\} = 0.64$
$\mu_{V'_m}(10) = max\{min(0.1, 0.1), min(0.3, 0.3), min(0.6, 0.5), min(0.64, 0.5)\} = max\{0.1, 0.3, 0.5, 0.5\} = 0.5$
$\mu_{V'_m}(15) = max\{min(0.1, 0.1), min(0.3, 0.3), min(0.6, 0.3), min(0.64, 0.3)\} = max\{0.1, 0.3, 0.3, 0.3\} = 0.3$

V	5	10	15
V'_m	0.64	0.5	0.3

6 (T) Razvijamo model strojnog učenja za predviđanje broja gledatelja na kinoprojekcijama. U obzir smo uzeli tri značajke: dan u tjednu, žanr filma i cijena produkcije filma. **Koji bi algoritam strojnog učenja bilo prikladno upotrijebiti za ovaj poblem, i zašto?**

A Neuronsku mrežu, jer predviđamo brojčanu vrijednost

B Neuronsku mrežu, jer je cijena produkcije filma brojčana značajka

C Naivan Bayesov klasifikator, jer predviđamo diskretne vrijednosti (cijeli brojevi)

D Stablo odluke, jer imamo tri značajke s podjednakom informacijskom dobiti

7 (R) Raspolažemo skupom primjera za *"Nezaboravno jadransko ljeto 2025., odmah nakon pandemije koronavirusa"*. Skup se sastoji od sljedećih primjera, svaki sa 4 značajke (Mjesto, Otok, Smještaj, Prijevoz) i ciljnom oznakom y:

i	Mjesto	Otok	Smještaj	Prijevoz	y
1	Istra	ne	privatni	auto	da
2	Istra	ne	privatni	avion	da
3	Dalmacija	da	hotel	auto	da
4	Dalmacija	da	hotel	bus	da
5	Kvarner	ne	kamp	bus	ne
6	Dalmacija	da	privatni	avion	ne
7	Istra	ne	kamp	auto	ne

Primijenite na ovaj skup primjera algoritam ID3. U slučaju da je u nekom koraku više značajki ima jednaku vrijednost informacijske dobiti, izaberite koja je u tablici navedena prva (ona ljevija). **Kako izgleda dobiveno stablo odluke?**

A Korijenski čvor stabla je Smještaj, a njegovo dijete je čvor Mjesto s informacijskom dobiti 0.251

B Korijenski čvor stabla je Mjesto, a njegovo dijete je čvor Smještaj s informacijskom dobiti 0.251

C Korijenski čvor stabla je Smještaj, a njegovo dijete je čvor Mjesto s informacijskom dobiti 0.918

D Korijenski čvor stabla je Mjesto, a njegovo dijete je čvor Smještaj s informacijskom dobiti 0.918

$$E(D) = -P(da)\log_2 P(da) - P(ne)\log_2 P(ne)$$

$$IG(D, x) = E(D) - \sum_{v \in V(x)} \frac{|D_{x=v}|}{|D|} E(D_{x=v})$$

$\langle m, n \rangle$ – skup sa m pozitivnih primjera i n negativnih primjera

Entropija cijelog skupa:

$$E(\langle 4, 3 \rangle) = -\frac{4}{7}\log_2\frac{4}{7} - \frac{3}{7}\log_2\frac{3}{7} = 0.985$$

značajka **Mjesto:**

Istra	Dalmacija	Kvarner
$E(\langle 2,1\rangle) = -\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3} = 0.918$	$E(\langle 2,1\rangle) = 0.918$	$E(\langle 0,1\rangle) = 0$

$$IG(D, Mjesto) = 0.985 - \frac{3}{7}\cdot 0.918 - \frac{3}{7}\cdot 0.918 = 0.198$$

značajka **Smještaj:**

Privatni	Hotel	Kamp
$E(\langle 2,1\rangle) = 0.918$	$E(\langle 2,0\rangle) = 0$	$E(\langle 0,2\rangle) = 0$

$$IG(D, Smještaj) = 0.985 - \frac{3}{7}\cdot 0.918 = 0.592$$

Veći IG, stoga je Smještaj korijenski čvor.
Treba još izračunati IG za Mjesto, koji se nalazi ispod brida Smještaj=privatni, pa kao entropiju početnog skupa uzimamo $E\left(D_{Smještaj=privatni}\right) = E(\langle 2,1\rangle) = 0.918$:

Istra	Dalmacija	Kvarner
$E(\langle 0,2\rangle) = 0$	$E(\langle 1,0\rangle) = 0$	$E(\langle 0,0\rangle) = 0$

$$IG\left(D_{Smještaj=privatni}, Mjesto\right) = 0.918$$

8 (P) Radimo svoju implementaciju algoritma ID3. Kako bismo spriječili da se model prenauči, implementirali smo i podrezivanje stabla na dubini k. Nažalost, kod implementacije funkcije informacijske dobiti (IG) potkrala nam se mala pogreška: zaboravili smo negirati vrijednost pri izračunu entropije $E(D)$ skupa primjera D. Dakle, umjesto da izračunava $E(D)$, naša implementacija izračunava $-E(D)$. Neka je M_1^k stablo odluke koje dobivamo učenjem takvim pogrešno implementiranim algoritmom ID3, podrezano na neku konačnu dubinu k. Neka je M_2^k stablo koje bismo dobili da smo algoritam ID3 implementirali ispravno i naučeno stablo podrezali na neku konačnu dubinu k. Ako $k = \infty$, onda to znači da stablo ne podrezujemo. Neka je $E_u(M)$ pogreška učenja modela M, a $E_p(M)$ pogreška modela M na skupu za provjeru. **Što od sljedećeg možemo očekivati da vrijedi?**

A $E_p(M_2^\infty) = E_u(M_2^\infty)$ C $E_p(M_1^k) > E_p(M_2^k)$
B $E_u(M_2^k) = E_u(M_2^\infty)$ D $E_u(M_2^k) < E_u(M_2^\infty)$

A. Nepodrezano stablo će sve primjere za učenje točno klasificirati, ali ne nužno i one za provjeru
B. Nepodrezano stablo će sve primjere za učenje točno klasificirati, tj. pogreška = 0.
Podrezano stablo neće sve primjere za učenje točno klasificirati, tj. pogreška > 0.
Na primjerima za učenje, pogreška podrezanog stabla će biti veća od pogreške nepodrezanog stabla
C. Ako se koristi kriva funkcija entropije, stablo će se krivo izgraditi i imati veću pogrešku nego pravilno izgrađeno stablo
D. Isto kao B

9 (R) Mali je Ivica svakog svakog ljeta u zadnjih sedam godina naučio jedan novi programski jezik. Svoja vrijedna iskustva sažeo je u listu *"Programski jezik koji mi se jako sviđa"*, gdje je svaki jezik opisao četirima značajkama, te je naznačio je li mu se dotični jezik svidio ($y = 1$) ili nije ($y = 0$). Ta lista izgleda ovako:

i	Evaluacija	Izvođenje	Paradigma	Provjera tipova	y
1	lijena	kompajler	imperativna	statička	0
2	striktna	interpreter	deklarativna	dinamička	0
3	lijena	kompajler	imperativna	dinamička	0
4	lijena	interpreter	hibridna	statička	0
5	striktna	interpreter	imperativna	statička	1
6	lijena	kompajler	hibridna	dinamička	1
7	striktna	kompajler	hibridna	dinamička	1

Ovog ljeta Mali Ivica želi puno jesti i spavati te opet naučiti novi programski jezik. U užem je izboru jezik **x** sa sljedećim karakteristikama: $\mathbf{x} = (\text{lijena}, \text{interpreter}, \text{hibridna}, \text{dinamička})$. Međutim, ovog bi puta Mali bi Ivica volio unaprijed znati hoće li mu se dotični programski jezik svidjeti, tako da ne gubi cijelo ljeto bezveze. Pomozite Malom Ivici te na gornji skup primjera primjenite naivan Bayesov klasifikator s Laplaceovim zaglađivanjem "dodaj jedan". **Koliko iznosi vjerojatnost da bi se Malom Ivici programski jezik x jako svidio?**

A 0.694   B 0.799   C 0.856   D 0.431

$$P(da) = \frac{3}{7}$$

$$P(ne) = \frac{4}{7}$$

Izračun MAP-hipoteze uz Laplaceovo zaglađivanje "dodaj jedan":

$$y = da: \quad P(da)P(lijena|da)P(interpreter|da)P(hibridna|da)P(dinamička|da)$$
$$= \frac{3}{7} \cdot \frac{1+1}{3+2} \cdot \frac{1+1}{3+2} \cdot \frac{2+1}{3+3} \cdot \frac{2+1}{3+2} = 0.02057$$

$$y = ne: \quad P(ne)P(lijena|ne)P(interpreter|ne)P(hibridna|ne)P(dinamička|ne)$$
$$= \frac{4}{7} \cdot \frac{3+1}{4+2} \cdot \frac{2+1}{4+2} \cdot \frac{1+1}{4+3} \cdot \frac{2+1}{4+2} = 0.02721$$

$$P(da|X) = \frac{0.02057}{0.02057 + 0.02721} = 0.431$$

10 (T) Naivan Bayesov klasifikator nazivamo "naivnim" jer model pretpostavlja uvjetnu nezavisnost značajki x_j unutar klase y. Uz tu pretpostavku, izglednost klase $P(x_1, \ldots, x_n|y)$ možemo zamijeniti umnoškom $\prod_{j=1}^{n} P(x_j|y)$. **Koja je motivacija za uvođenje pretpostavke uvjetne nezavisnosti?**

A Mogućnost korištenja značajki koje nisu binarne

B Veća točnost modela na skupu za učenje

C Mogućnost generalizacije na neviđene primjerke

D Mogućnost klasifikacije u više od dvije klase

Bez uvjetne nezavisnosti, možemo računati izglednost isključivo primjera čije vjerojatnosti $P(x_1, \ldots, x_n|y)$ imamo. Pošto želimo model koji može generalizirati tj. računati izglednost klase za primjere tj. kombinacije značajki čije vjerojatnosti $P(x_1, \ldots, x_n|y)$ nemamo, uvodi se uvjetna nezavisnost.

11 (R) Unaprijednu potpuno povezanu slojevitu neuronsku mrežu arhitekture $3 \times 2 \times 2$ sa sigmoidnim prijenosnim funkcijama učimo preslikavanje $R^3 \to R^2$, odnosno skup primjeraka za učenje sadrži zapise oblika $(x_1, x_2, x_3) \mapsto (y_1, y_2)$. Trenutačne vrijednosti težina su:

$$w_{0,1}^{(1)} = -1, w_{1,1}^{(1)} = 0.1, w_{2,1}^{(1)} = 1, w_{3,1}^{(1)} = 1, w_{0,2}^{(1)} = 0.5, w_{1,2}^{(1)} = 0.4, w_{2,2}^{(1)} = -2, w_{3,2}^{(1)} = 0.8,$$
$$w_{0,1}^{(2)} = -0.4, w_{1,1}^{(2)} = -2, w_{2,1}^{(2)} = 1, w_{0,2}^{(2)} = 0.4, w_{1,2}^{(2)} = 1, w_{2,2}^{(2)} = 0.3.$$

Primjerak koji trenutačno razmatramo je $(0.2, -0.1, 0.2) \mapsto (1, 0)$. Učenje mreže provodi se postupkom propagacije pogreške unazad na temelju pojedinačnih primjeraka. Neka je iznos stope učenja jednak 10. Provedite postupak učenja za dani primjerak. **Koliko iznosi zbroj $w_{1,2}^{(1)} + w_{3,1}^{(1)}$ nakon provedenih korekcija?** (Odgovori su zaokruženi na četiri decimale.)

A 1.3137 B 1.3521 C 1.2627 D 1.4752

$$sigm(x) = \frac{1}{1 + e^{-x}}$$

Računanje izlaza neurona:

$$y_1^{(2)} = sigm(0.2 * 0.1 + (-0.1) * 1 + 0.2 * 1 - 1) = sigm(-0.88) = 0.293$$
$$y_2^{(2)} = sigm(0.2 * 0.4 + (-0.1) * (-2) + 0.2 * 0.8 + 0.5) = sigm(0.94) = 0.719$$
$$y_1^{(3)} = sigm(0.293 * (-2) + 0.719 * 1 - 0.4) = sigm(-0.267) = 0.434$$
$$y_2^{(3)} = sigm(0.293 * 1 + 0.719 * 0.3 + 0.4) = sigm(0.909) = 0.713$$

Izračun pogrešaka neurona:

Pogreške izlaznog sloja: $\delta_j = o_j * (1 - o_j) * (t_j - o_j)$

$$\delta_1^{(3)} = 0.434 * (1 - 0.434) * (1 - 0.434) = 0.139$$
$$\delta_2^{(3)} = 0.713 * (1 - 0.713) * (0 - 0.713) = -0.146$$

Pogreške u skrivenom sloju: $\delta_j^{(k)} = y_j^{(k)} * \left(1 - y_j^{(k)}\right) * \left(w_{j,1}^{(k)} * \delta_1^{(k+1)} + \cdots + w_{j,m}^{(k)} * \delta_m^{(k+1)}\right)$

$$\delta_1^{(2)} = 0.293 * (1 - 0.293) * \left(-2 * 0.139 + 1 * (-0.146)\right) = -0.088$$
$$\delta_2^{(2)} = 0.719 * (1 - 0.719) * \left(1 * 0.139 + 0.3 * (-0.146)\right) = 0.019$$

Podešavanje težina: $w_{i,j}^k = w_{i,j}^k + \eta y_i^{(k)} \delta_j^{(k+1)}$

$$w_{1,2}^{(1)} = 0.4 + 10 * 0.2 * 0.019 = 0.438$$
$$w_{3,1}^{(1)} = 1 + 10*0.2*(-0.088) = 0.824$$

$$w_{1,2}^{(1)} + w_{3,1}^{(1)} = 1.262$$

12 (T) McCulloch-Pittsov model umjetnog neurona nastao je davne 1943. godine. **Što od sljedećega *nije* točno za McCulloch-Pitsov model?**

A Neuron nema memoriju, odnosno uvijek će za isti ulaz dati isti izlaz

B Tijelo neurona s ulaznim dendritima modelirano je težinskom sumom

C Brzina širenja električkih impulsa modelirana je težinama

D Funkcioniranje aksonskog vlakna modelirano je prijenosnom funkcijom

13 (P) Raspolažemo dvama skupovima primjeraka za učenje (iste su dimenzionalnosti ulaza) za binarni klasifikacijski problem. Kao klasifikator koristimo TLU-perceptron. Poznato je da nad oba skupa Rosenblattov algoritam uspješno dolazi do rješenja. Neka mu nad prvim skupom za to treba n_1 koraka, a nad drugim skupom n_2 koraka. Razmotrite sada treći skup primjeraka za učenje koji predstavlja uniju prethodnih dvaju. **Koliko će koraka trebati Rosenblattovu algoritmu da nauči ispravno klasificirati taj skup?**

- A $max(n_1, n_2)$ koraka
- C n_1 ili n_2 koraka, ovisno o inicijalizaciji težina
- B Nemamo garancije da će učenje biti uspješno
- D Minimalno n_1 koraka

Recimo da je prvi skup $\{ (1,1) \rightarrow 1, (1,2) \rightarrow -1 \}$ i drugi skup $\{ (2,1) \rightarrow -1, (2,2) \rightarrow 1 \}$
TLU-perceptron će svaki skup zasebno uspjeti naučiti jer može povući linearnu decizijsku granicu u koordinatnom sustavu između primjera klasificiranog sa 1 i onog sa -1.
TLU-perceptron ne može ispravno naučiti uniju tih dvaju skupova jer se takva granica između primjera ne može naći, tj. razredi su linearno-nerazdvojivi.

14 (T) Kroz povijest razvoja područja umjetne inteligencije izmjenjivali su se periodi snažnog razvoja te periodi stagnacije (tzv. "zime"). **Što je dovelo do okončanja zime umjetne inteligencije pretkraj 80-ih godina?**

- A Otkriće algoritma propagacije pogreške unatrag
- B Otkriće prijenosnih funkcija poput zglobnice
- C Otkriće Rosenblattovog algoritma učenja perceptrona
- D Dostupnost grafičkih kartica za ubrzavanje računanja

15 (R) Skup primjera za učenje $\{(x_2, x_1, y)\}$ je $\{(1,1,-1),(2,4,1),(1,2,-1),(3,3,1),(2,1,-1),(4,2,1)\}$. Učenje se provodi uporabom perceptrona TLU s izlaznim vrijednostima -1 i 1 te stopom učenja $\eta = 1$. Početne vrijednosti težinskih faktora su $(w_2, w_1, w_0) = (1.3, 1.2, -3.2)$. Provedite postupak učenja Rosenblattovim algoritmom. **Koliko se puta tijekom učenja provode korekcije težina te koje su njihove konačne vrijednosti?**

- A 3 puta, $(1.3, 1.2, -5.2)$
- C 4 puta, $(5.5, -1.5, 10)$
- B 4 puta, $(1.3, -2.5, 12)$
- D postupak ne konvergira

Učenje: $\mathrm{w_i} \leftarrow \mathrm{w_i} + \eta(t - o)\mathrm{x_i}$

$1.\ y = step(1 * 1.3 + 1 * 1.2 - 3.2) = -1$
$2.\ y = step(2 * 1.3 + 4 * 1.2 - 3.2) = 1$
$3.\ y = step(1 * 1.3 + 2 * 1.2 - 3.2) = 1$
$\mathrm{w_2} = 1.3 + (-2) * 1 = -0.7$
$\mathrm{w_1} = 1.2 + (-2) * 2 = -2.8$
$\mathrm{w_0} = -3.2 + (-2) * 1 = -5.2$

$4.\ y = step(3 * (-0.7) + 3 * (-2.8) - 5.2) = -1$
$\mathrm{w_2} = -0.7 + 2 * 3 = 5.3$
$\mathrm{w_1} = -2.8 + 2 * 3 = 3.2$
$\mathrm{w_0} = -5.2 + 2 * 1 = -3.2$

$5.\ y = step(2 * 5.3 + 1 * 3.2 - 3.2) = 1$
$\mathrm{w_2} = 5.3 + (-2) * 2 = 1.3$
$\mathrm{w_1} = 3.2 + (-2) * 1 = 1.2$
$\mathrm{w_0} = -3.2 + (-2) * 1 = -5.2$

6. $y = step(4 * 1.3 + 2 * 1.2 - 5.2) = 1$

2. epoha

1. $y = step(1 * 1.3 + 1 * 1.2 - 5.2) = -1$
2. $y = step(2 * 1.3 + 4 * 1.2 - 5.2) = 1$
3. $y = step(1 * 1.3 + 2 * 1.2 - 5.2) = -1$
4. $y = step(3 * 1.3 + 3 * 1.2 - 5.2) = 1$
5. $\mathrm{y} = \mathrm{step}(2 * 1.3 + 1 * 1.2 - 5.2) = -1$

Sada je sve je točno klasificirano.

16 (P) Ako kod algoritma Ant System postavimo τ_0 na vrijednost koja je puno manja od količine feromonskih tragova koju deponira jedan mrav, **što će biti posljedica?**

A isparavanje feromonskih tragova će biti preveliko

B algoritam će dugo vremena istraživati nasumične puteve, prije no što ga mravi uspiju fokusirati

C algoritam će rapidno konvergirati prema globalnom optimumu

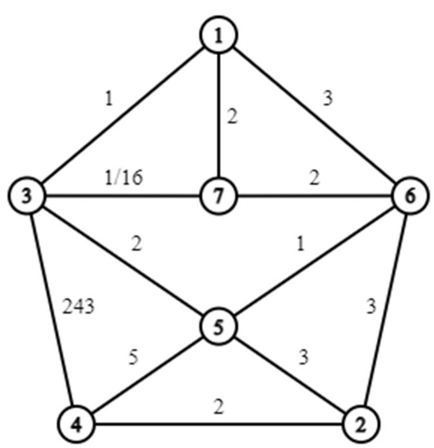
D algoritam će izgubiti mogućnost istraživanja kvalitetnih rješenja

A. τ_0 nema nikakve veze s isparavanjem
B. Ne, to će se desiti ako je τ_0 puno veći od količine feromona koju deponiraju mravi
C. Algoritam će brzo konvergirati, ali ne prema globalnom optimumu
D. Da, mravi će zaglaviti u lokalnom optimumu i neće istraživati druge staze

17 (R) Uporabom mravlje kolonije traži se ciklus kroz graf. Poznati su sljedeći podatci: $\tau_{1,3} = 0.5$, $\tau_{1,6} = \frac{1}{\sqrt{3}}$, $\tau_{1,7} = 2$, $\tau_{2,4} = 1$, $\tau_{2,5} = \sqrt{3}$, $\tau_{2,6} = \frac{1}{3}$, $\tau_{3,4} = 3$, $\tau_{3,5} = 10\sqrt{2}$, $\tau_{3,7} = 0.5$, $\tau_{4,5} = 10$, $\tau_{5,6} = 2$, $\tau_{6,7} = \frac{1}{2}$. $\eta_{1,3} = 2$, $\eta_{1,6} = 3$, $\eta_{1,7} = \frac{1}{\sqrt{2}}$, $\eta_{2,4} = \sqrt{2}$, $\eta_{2,5} = 1$, $\eta_{2,6} = 3\sqrt{3}$, $\eta_{3,4} = 3\sqrt{3}$, $\eta_{3,5} = 0.1$, $\eta_{3,7} = 0.5$, $\eta_{4,5} = \frac{1}{2\sqrt{5}}$, $\eta_{5,6} = 0.5$, $\eta_{6,7} = 2\sqrt{2}$. Također, sve dane vrijednosti su simetrične, tj. $\tau_{i,j} = \tau_{j,i}$ i $\eta_{i,j} = \eta_{j,i}$. Dodatno, $\alpha = 2$ i $\beta = 2$. Prvi mrav kreće iz čvora 1. Kada mrav treba donijeti vjerojatnosnu odluku, pretpostavite da će ishod slučajnog odabira odgovarati najvjerojatnijem. Ako iz nekog čvora mrav ne može dalje, konstrukcija ciklusa se prekida. Uz te pretpostavke odredite ciklus koji će taj mrav konstruirati. **Slijed od koja tri čvora je dio tog ciklusa?**

A 3, 7, 6 B 5, 3, 1 C 2, 5, 4 D Mrav neće uspjeti konstruirati ciklus

Pošto će ovdje mrav uvijek ići onim bridom čija je vjerojatnost odabira najveća, ne moramo računati vjerojatnosti već se u svakom koraku ide bridom čiji je umnožak $\tau^\alpha\eta^\beta$ najveći.



Prvo je potrebno nacrtati graf. Pošto je $\alpha = \beta = 2$, uz svaki brid je oznaka koja predstavlja umnožak $\tau^2\eta^2$za taj brid

Mrav svaki čvor može posjetiti samo jedan put, osim čvora 1 iz kojeg krećemo i u koji se trebamo vratiti.

Konstruirati će se ciklus 1 → 6 → 2 → 5 → 4 → 3 → 1

18 (T) Problemi zadovoljavanja ograničenja (engl. *constraint satisfaction problem*, CSP) podskupina su problema pretraživanja prostora stanja. **Što je karakteristično za ovu vrstu problema?**

A Rješenje omogućava rekonstrukciju puta do početnog stanja

B Bitno nam je samo konačno stanje

C Mogu se rješavati algoritmom A*, ali samo uz optimističnu heuristiku i skup posjećenih stanja

D Ispitni predikat definiramo tako da uspoređuje stanje sa zadanim predloškom (kao kod slagalice)

19 (R) Minimum funkcije $f(x, y, z) = (x+1)^2+(y-3)^2+(z+2)^2+xz$. pronalazi se genetskim algoritmom ostvarenim u obliku jednostavne troturnirske selekcije. Svaka varijabla je pri tome ostvarena s 2 bita te se pretražuje cjelobrojno područje: x iz intervala $[0, 3]$, a y i z iz intervala $[-1, 2]$. U kromosomu najprije dolaze bitovi od x, pa od y, pa od z. U jednom koraku odabrana su sljedeća tri kromosoma: K_1 = 000101, K_2 = 100001, K_3 = 001011. Kao funkcija dobrote koristi se negirana funkcija f, dakle dobrota od (x, y, z) je jednaka $-f(x, y, z)$. Koristi se križanje s jednom točkom prijeloma na polovici kromosoma. **Izračunajte dobrotu djeteta koje će biti vraćeno u populaciju.** Pretpostavite da prilikom križanja mutacija uvijek promijeni zadnji bit kromosoma (onaj najdesniji). Ako operator križanja generira više djece, u populaciju će se vratiti najbolje od generirane djece.

A -3 B -21 C -26 D -29

$$f(x,y,z) = (x+1)^2 + (y-3)^2 + (z+2)^2 + xz$$

Za određivanje vrijednosti varijable u intervalu $[x_{\min}, x_{\max}]$ iz binarnog koda s n bitova:

$$x = x_{\min} + \frac{k}{2^n\text{-}1} \cdot (x_{\max} - x_{\min})$$

npr. $y(00) = -1 + \frac{0}{2^2\text{-}1} \cdot \big(2-(-1)\big) = -1$, $y(01) = -1 + \frac{1}{2^2\text{-}1} \cdot \big(2-(-1)\big) = 0$ itd…

	x	y	z
00	0	-1	-1
01	1	0	0
10	2	1	1
11	3	2	2

$K_1 = 000101 \Rightarrow x = 0, y = 0, z = 0,\quad Dobrota = -f(0,0,0) = -14$
$K_2 = 100001 \Rightarrow x = 2, y = -1, z = 0,\quad Dobrota = -f(2,-1,0) = -29$
$K_3 = 001011 \Rightarrow x = 0, y = 1, z = 2,\quad Dobrota = -f(0,1,2) = -21$

K_2 ima najmanju dobrotu, stoga je eliminirano i križaju se kromosomi K_1 i K_3:
$K_1 = 000101$
$K_3 = 001011$

Nastaju dva djeteta + dodana mutacija:
$C_1 = 000010$
$C_2 = 001100$

Dekodiramo vrijednosti varijabli x,y,z i računamo dobrotu djece:

$$-f(0,-1,1) = -26$$
$$-f(0,2,-1) = -3$$

Drugo dijete ima najbolju dobrotu -3, stoga će se to dijete vratiti u populaciju.

20 (P) Genetskim algoritmom tražimo maksimum funkcije $f(x, y, z)$, koristeći binarnu reprezentaciju rješenja. Vrijednost svake od varijabli pretražuje se u intervalu $[10, 90]$, pri čemu je potrebno osigurati da se to pretraživanje provodi barem s preciznošću 0.001. **Od koliko se *minimalno* bitova treba sastojati kromosom?**

A 49 B 16 C 48 D 51

Interval veličine 80, uz preciznost barem 0.001, pa svaka varijabla ima barem $80 * 1000 = 80000$ mogućih vrijednosti. Tu varijablu možemo predstaviti 17-bitnim binarnim brojem jer je 2^{17} veće od 80000. Tri su varijable: x, y, z, stoga nam treba 51 bit.

Napravio @PaleAle