10. fejezet

Mesterséges intelligencia

MI problémák és az útkeresési feladat kapcsolata. Állapottér-reprezentáció. Keresések – lokális keresés, visszalépéses keresés, gráfkeresés – jellemzése. Kétszemélyes játékok. Evolúciós algoritmusok. Mesterséges neuronhálózatok. Automatikus logikai következtetés.

10.1. MI problémák

A mesterséges intelligencia nem pontosan definiált, körülhatárolt tudományterület. (kezdeti célja az emberi gondolkodás reprodukálása volt, ám ez túl nagyra törőnek bizonyult). Szemléletesen a mesterséges intelligencia módszereivel megoldott feladatok általában:

- emberi szemmel tekintve nehezek,
- az algoritmuson túl valamilyen speciális ismeretet, tudást, gyakorlatot, intuíciót igényelnek.

Ennél fogva a MI módszerei az emberi jellegű, intelligens gondolkodás reprodukálása szempontjából hasznos technikákon, elveken nyugszanak.

10.2. Az útkeresési feladat

A mesterséges intelligencia problémáit gyakran kezelhetjük egy gráfban való útkeresési feladatként, ahol a *problématér* elemein, azaz a feladatra adható lehetséges megoldások egy halmazán szomszédossági kapcsolatokat definiálunk, és ezen kapcsolatok mentén keressük a *helyes megoldás*t egy rögzített *kiinduló állapot*ból. Speciális esetben a szomszédossági viszonyok valamilyen

*elemi transzformációs lépések*et fejeznek ki az állapotok között, ekkor kereshetünk a helyes válaszhoz vezető utat, mely így a megoldás előállításának lépéseit reprezentálja.

Alapvető gráf-fogalmak. Él, utód, szülő, élköltség, út, út hossza, út költsége, út optimális költsége, optimális költségű út.

10.2.1. Definíció (δ-gráf). *Olyan élsúlyozott ($c : A \to \mathbb{R}$), irányított gráf, melyre teljesül a*

- *σ-tulajdonság: $\exists \sigma \in \mathbb{N}$, hogy a gráf minden n csúcsára $|\Gamma(n)| \leq \sigma$ ($\Gamma(n)$ az n csúcs szomszédainak halmaza),*
- *δ-tulajdonság: a gráf minden élére a c költségfüggvény értéke nem kisebb, mint egy rögzített $\delta \in \mathbb{R}^+$.*

10.2.2. Definíció (Gráfreprezentáció). *Egy útkeresési probléma gráfreprezentációja egy irányított (és élsúlyozott) δ-gráfból (N, A, c), egy $s \in N$ startcsúcsból és a $T \subset A$ célcsúcsok halmazából áll.*

10.2.3. Definíció (Kereső rendszer). *Egy kereső rendszer részei:*

- globális munkaterület: *a keresés során megszerzett és megőrzött információk halmaza,*
- szabályok: *a globális munkaterületet változtató operátorok/függvények,*
- vezérlési stratégia: *a végrehajtandó szabályt kiválasztó módszerek, melyeket három csoportra bontunk:*
  1. elsődleges stratégiák: *általános, a feladattól független elvek,*
  2. másodlagos stratégiák: *a probléma adott reprezentációjától függő módszerek,*
  3. heurisztikák: *a konkrét feladat ismeretét felhasználó, speciális technikák.*

10.3. Állapottér-reprezentáció

10.3.1. Definíció (Állapottér-reprezentáció). *Egy probléma állapottér-reprezentációja négy részből áll:*

- állapottér: *a feladat homlokterében álló adat lehetséges értékei,*
- Műveletek: *állapotból állapotba vezető műveletek – általában egy előfeltételből és az annak teljesülése esetén a globális munkaterületen elvégzendő transzformáció leírásából állnak,*
- kezdő állapot: *a megoldási folyamat kiindulópontja,*
- célállapotok: *az elérendő állapotok halmaza.*

A probléma megoldása *műveletek sorozata, melyek a kezdő állapotból egy célállapotba vezetnek.*

Megjegyzés: *a problématér itt a lehetséges műveletsorozatok halmaza, mely nyilván nem azonos az állapottérrel.*

10.3.2. Definíció (Állapotgráf). *Az állapottér-reprezentáció állapotgráfja egy olyan irányított gráf, melynek csúcsai a állapottér elemei (állapotok), és akkor vezet él egy csúcsból egy másikba, ha a két állapot között megvalósítható átmenet valamelyik művelet segítségével.*

10.3.3. Definíció (Keresési tér). *A keresés általában az állapotgráf folyamatos, keresés közbeni építésével zajlik. Ennek folyamán az állapotgráf torzulhat, pl. ha nem fordítunk gondot a körök felfedezésére. Így a keresési tér az a gráf, ahol a keresés „valójában" keres.*

10.3.1. Megjegyzés

1. *Modellek különbözősége:* ugyanannak a problémának sok állapottér-reprezentációs modellje képzelhető el. Általában hasznos azt a modellt választani, amely kevesebb állapotot tartalmaz.

10.4. Keresések

10.4.1. Lokális keresések

Hegymászó módszer

A globális munkaterületen mindig csak az aktuális csúcsot tároljuk, illetve azt a csúcsot, ahonnak ide jutottunk. Egy *célfüggvény* (heurisztika) segítségével minden csúcshoz értéket rendelünk, mely maximumát a célcsúcsokban veszi fel. Minden lépésben a szomszédos csúcsok közül a legnagyobb, az aktuális csúcsnál nagyobb értékű csúcsra „lépünk".

Hátrányok. Csak erős heurisztika esetén sikeres, mivel:

- lokális maximum körül, és ekvidisztans felületen eltévedhet,
- zsákutcába beragadhat.

Tabu-keresés

A hegymászó módszer javítása, melyben a globális munkaterületen néhány korábbi állapotot is nyilvántartunk, és ezekre nem lépünk vissza. Nyilvántartjuk továbbá az eddig legjobbnak bizonyult elemet. A keresés akkor *teminál*, ha az optimális csúcs, illetve annak célfüggvény-értéke sokáig nem változik – vagy célcsúcsba értünk.

Hátrányok.

- a tabu-halmaz méretének meghatározására nincs általános elv,
- beragadhat, ha csak tabu-beli elemekre tudna lépni.

Szimulált hűtés

Olyan lokális keresés, ahol a következő elem választása véletlenszerű, és megengedünk esetleg „rosszabb" elemre való lépést is, a különbséggel fordítottan arányos valószínűséggel (pl. ha $e^{\frac{f(r)-f(n)}{T_k}} \geq random[0,1]$).

Finomhangolás. Az algoritmus finomhangolható a T_k érték folyamatos változtatásával: ha T_k csökken, az csökkenti a rosszabb elem kiválasztásának valószínűségét (*hűtési ütemterv*).

Értékelés. Aszimptotikusan konvergál egy optimális megoldáshoz, *de* annak véges lépésen belüli eléréséhez megfelelő hűtési ütemterv (heurisztika) szükséges.

10.4.2. Visszalépéses keresés

Felépítés

- *globális munkaterület:* egy startcsúcsból kiinduló út,
- *szabályok:* az út bővítése egy új csúccsal / visszalépés,

- *elsődleges vezérlési stratégia:* visszalépést csak végszükség esetén hajtunk végre.

Visszalépés feltételei

- zsákutca,
- zsákutca-torkolat (olyan csúcs, ahonnan már minden kivezető utat kipróbáltunk),
- kör,
- mélységi korlát.

Heurisztikák

- *sorrendi hurisztika:* a továbblépés irányának kiválasztása a lehetőségek rangsorolásával,
- *vágó heurisztika:* a lehetőségek közül a nem kellően igéretesek elhagyása.

Előnyök

- a munkaterület mérete (így a memóriaigény) viszonylag kicsi, a mélységi korláttal is korlátozható,
- könnyen implementálható (pl. rekurzív algoritmusként),
- ha a mélységi korláton belül van megoldás, azt mindig megtalálja.

Hátrányok

- ha a mélységi korláton belül nincs megoldás, akkor az algoritmus megoldás nélkül terminál,
- végtelen gráfban a terminálás nem biztosított,
- véges gráfban mindig terminál, de nem feltétlenül talál optimális megoldást (enne érdekében iterációba szervezhető),
- egy „rossz” lépés egy nagy részgráf felesleges felderítését hozza magával,
- mivel a korábban bejárt utakat nem tartjuk számon, zsákutcák többszöri bejárása előfordulhat.

10.4.3. Gráfkeresés

Felépítés

- *globális munkaterület:* a keresési tér már feltárt részgráfja,
- *szabályok:* egy csúcs kiterjesztése (lezárása, és utódainak felvétele a *nyílt* halmazba),
- *elsődleges vezérlési stratégia:* a legkedvezőbb csúcs kiterjesztésére törekszik a kiértékelő függvény ($f : N \to \mathbb{R}$),
- *terminálási feltétel:* a munkaterületen megjelenik egy célcsúcs, vagy megakad az algoritmus.

Az algoritmus tehát kezdetben a startcsúcsot tartalmazza a *nyílt* halmazban, és minden lépésben kiterjeszti a legjobb f-értékű csúcsot a *nyílt* halmazból. Nyilván szokás tartani egy *optimális feszítőfát* a feltárt részgráfban, mégpedig úgy, hogy minden csúcshoz tároljuk a *szülő csúcsot* ($\pi(n)$) és az *odavezető út költségét* ($g(n)$).

Körök kezelése

Konzisztenssé tétel

Ha egy csúcshoz jobb utat találtunk, akkor a teljes feltárt részgráfon frissíthetjük az információkat. Ez műveletigényes megoldás.

Visszahelyezés

Ha egy csúcshoz jobb utat találtunk, akkor helyezzük vissza a csúcsot a *nyílt* halmazba! Így, ha az algoritmus újra kiterjeszti, a konzisztencia fokozatosan helyreáll.

Tulajdonságok

10.4.1. Tétel. *A gráfkeresés működése során egy csúcsot legfeljebb véges sokszor terjeszt ki.*

10.4.2. Tétel. *A gráfkeresés véges reprezentációs gráfban mindig terminál, és talál megoldást, ha létezik.*

10.4.3. Tétel (Invariáns tulajdonság). *Legyen n egy tetszőleges, s-ből elérhető csúcs. A gráfkeresés az n csúcs kiterjesztése előtt bármelyik $s \to n$ optimális úton mindig nyilvántart egy olyan m csúcsot, amelyikre teljesül, hogy*

- $m \in nylt$,
- $g(m) = g^*(m)$
- *minden m csúcsot megelőző csúcs végleg zárt, azaz minden ilyen k csúcsra $g(k) = g^*(k)$.*

Függvények

- g és g^* – a startcsúcsból n-be vezető út aktuális és optimális költsége,
- h és h^* – az n-ből egy célcsúcsba vezető út becsült és optimális költsége (heurisztikus függvény),
- f és f^* – a kiértékelő függvény, illetve az n csúcson át célba vezető, legkisebb költségű út költsége.

Neminformált keresések

mélységi keresés	$f \equiv -g, c \equiv 1$	végtelen gráfokban csak mélységi korláttal garantált a megoldás
szélességi keresés	$f \equiv g, c \equiv 1$	optimális megoldást ad, egy csúcsot legfeljebb egyszer terjeszt ki
egyenletes keresés	$f \equiv g$	optimális megoldást ad, egy csúcsot legfeljebb egyszer terjeszt ki

Heurisztikus keresések

előre tekintő (mohó) keresés	$f = h$	szeszélyes, eredményessége erősen függ a heurisztikus függvénytől
A algoritmus	$f = g + h$ és $h \geq 0$	mindig talál megoldást, ha létezik
A^* algoritmus	A algoritmus, $h \leq h^*$ (megengedhető) és $h(t) = 0$	mindig optimális megoldást talál
A^c (következetes) algoritmus	A algoritmus, $h(n) - h(m) \leq c(n,m)$ (monoton megszorításos)	mindig optimális megoldást talál, és egy csúcsot legfeljebb egyszer terjeszt ki
B algoritmus	A algoritmus, F aktuális küszöbérték, HA $min_f(nylt) < F$, akkor egyenletes keresést használunk	optimális megoldást talál, és egy árkon belül minden csúcsot legfeljebb egyszer terjeszt ki ($\mathcal{O}(n^2)$)
B' algoritmus	változó heurisztikájú B algoritmus, mely folyamatosan közelebb viszi a heurisztikus függvényt a monoton megszorításhoz	

Megjegyzések.

Monoton megszorításos heurisztika. Előállítása nehéz lehet, ezért az A^* algoritmus a leggyakoribb.

Algoritmusok összehasonlítása.

10.4.4. Definíció. *Egy h_1 heurisztikus függvény jobban informált h_2-nél, ha $h_1, h_2 \geq 0$ és $\forall n \in N : h_1 < h_2$.*

10.4.5. Tétel. *Monoton megszorításos és célban nulla heurisztikájú feladatok esetén az* A *algoritmus jobb az összes megengedhető algoritmusnál.*

10.4.6. Tétel. *A megengedhető algoritmusok között nincs jobb az* A *algoritmusnál.*

10.5. Kétszemélyes, véges, teljes információjú játékok

Kétszemélyes, teljes információjú játékok azok, melyekben két játékos felváltva lép (módosít a játék állásán meghatározott szabályok szerint), mindkét játékos a játék folyamán végig tisztában van a jelenlegi állás minden jellemzőjével, és a játékban semmilyen módon nem játszik szerepet a szerencse. A játék zéró összegű, ha amennyit az egyik játékos nyer, annyit veszít a másik.

Megköveteljük még, hogy a játékok véges lépésben véget érjenek, és a végső állásban vagy az egyik játékos nyerjen, vagy az állás döntetlen legyen.

Példák. Grundy mama játéka, Othello, Tic-tac-toe, boroskancsó-játék, ...

10.5.1. Definíció (Játékfa). *Olyan (*és/vagy-*) fa, melynek gyökerében a kezdő állás, a következő szinteken pedig váltakozva az egyik és a másik játékos lépései által létrehozható állások szerepelnek.*

És/vagy-*fát akkor használunk, ha az egyik játékos szemszögéből vizsgáljuk a játékot. Ekkor az adott (pl.* A*) játékos lehetőségei* vagy*, a másikéi (*B*)* és *kapcsolatban vannak.*

10.5.2. Definíció (Nyerő/nem vesztő stratégia). *Az* A *játékosnak van nyerő/nem vesztő stratégiája, ha a játék teljes,* A *szemszögéből elkészített* és/vagy-*fájában van olyan hiperút a kezdőállástól a levelekig, melynek minden levele az* A *számára kedvező (nyerő/döntetlen) állást reprezentál.*

10.5.3. Tétel (Nyerő/nem vesztő stratégia létezése). *Kétszemélyes, véges, teljes információjú játékban az egyik játékosnak mindig van nyerő/nem vesztő stratégiája.*

10.5.1. A játékfa részleges kiértékelése

Általában nem érdemes a teljes játékfát kiértékelni a számítási igény miatt. Ehelyett a várhatóan legkedvezőbb lépés meghatározása érdekében egy *statikus kiértékelő függvényt* definiálunk minden játékfa-csúcsra. Ennek segítségével próbáljuk az aktuális állásól kiinduló kis mélységű kereséssel meghatározni a legkedvezőbbnek tűnő lépést. *Legyen a továbbiakban* max *annak a játékosnak a neve, akinek a szemszögéből az állást vizsgáljuk, és* min *az ellenfélé!*

Minimax algoritmus

- Elkészítjük a játékfát rögzített mélységig,
- a levelekre meghatározzuk a kiértékelő függvény értékét,
- a kiszámolt értékeket a gráf élei mentén „felfuttatjuk" a gyökérelemig, *vagy* kapcsolat esetén maximumot, *és* esetén minimumot képezve a gyermekek értékeiből,
- végül a legfelső (*vagy*) szinten arra lépünk, amerre a legnagyobb függvényértékű gyerekcsúcs található.

Negamax algoritmus

A *minimax algoritmus* technikailag módosított változata, ahol a váltogatott maximum-minimumkeresés helyett minden szinten a függvényértékek -1-szeresét képezzük.

Átlagoló kiértékelés

Szintén a *minimax* algoritmus módosítása, ahol nem a maximumot/minimumot, hanem a néhány legnagyobb/legkisebb függvényértékű gyermekcsúcs átlagát futtatjuk fel a szülőre.

Ez az algoritmus a legjobbnak látszó állapot helyett inkább egy kevésbé jó függvényértékű, de biztonságos részfa felé mozdítja a játékost. Ezzel kijavíthatók a statikus kiértékelő függvény esetleges túlzott „optimizmusai".

Alfa-béta kiértékelés

A függvényértékek felfuttatására általában minimax algoritmust használó, a memóriaigényt és a számításigényt a biztosan felesleges kiértékelések elhagyásával csökkentő módszer.

Lényege, hogy ha egy szinten egy függvényértéket kiszámítottunk, akkor ez megadja nekünk a következő szinten kialakuló függvényérték egy alsó vagy felső korlátját. Ha egy másik ágon csak ezen a korláton kívül eső érték érkezhetne, akkor az ág további vizsgálatát elhagyhatjuk.

α	*max* szintjén jelenik meg, a lehetséges maximumérték alsó korlátja
β	*min* szintjén jelenik meg, a lehetséges minimumérték felső korlátja
a szabály	**vágunk, ha** $\alpha \geq \beta$

Változó mélységű kiértékelés

Célja, hogy a kiértékelő függvény minden ágon reális értéket mutasson.

Szabály. Egy adott mélységtől kezdve akkor vesszük be egy csúcs utódait a részfába, ha nem teljesül a *nyugalmi teszt* ($|f(szulo) - f(gyerek)| < K$).

10.6. Evolúciós algoritmusok

Felépítés

- *globális munkaterület:* megoldások egy lehetséges halmaza (populáció),
- *szabályok:* szelekciós, rekombinációs, mutációs és visszahelyező operátorok,
- *elsődleges vezérlési stratégia:* a populációra a fenti sorrendben mind a négy operátort alkalmazzuk, így új populációhoz jutunk.

10.6.1. Definíció (Fitness-függvény). *Egy állapot (*egyed*) rátermettségét jellemző érték, mely várhatóan annál nagyobb, minél közelebb áll az egyed a keresett eredményhez.*

10.6.2. Definíció (Kromoszóma). *Egy egyed állapotát leíró jelsorozat.*

Szelekciós operátorok. A szelekció célja a legrátermettebb egyedek kiválasztása keresztezésre, a kevésbé rátermett egyedek számára is esélyt biztosítva (a rejtőzködő hasznos tulajdonságok elvesztésének megakadályozására és a variabilitás fenntartására).

Rátermettséggel arányos rulettkerék-algoritmus,

Rangsorolás ban elfoglalt hellyel arányos,

Versengés (véletlen egyedcsoportok legjobb egyedei) alapján,

Csonkolás a rátermettség szerinti legjobb néhány egyedből.

Rekombinációs operátorok.

Egy- és többpontos keresztezés esetén ködszakaszokat cserélünk a kromoszómában,

Egyenletes keresztezés kor kódelemeket cserélünk egymással,

Permutációk keresztezése pl. ciklikusan haladva egy kiválasztott elem cseréjétől kezdve,

Köztes rekombináció (tömbökre) esetén a szülők által kifeszített hipertégla vagy testátlójának egy eleme lesz az utód.

Mutáció. Egy utód kis mértékű, véletlen módosítása.

Visszahelyezés. Célja a régi populációból és az utódokból az új populáció kialakítása. Jellemző értékei:

- utodkepzesi rata $= \frac{\text{utodok szama}}{\text{populacio egyedszama}}$
- visszahelyezesi rata $= \frac{\text{lecserelt egyedek szama}}{\text{populacio egyedszama}}$

10.7. Mesterséges neuronhálózatok

Felhasználási területei például az asszociatív tanulás, optimalizálás, közelítés, osztályozási feladatok.

Felépítés

- *mesterséges neuron:* bemenő értékekből kimenő értéket számoló, esetleg kis belső memóriával rendelkező egység, melynek számítási képlete változtatható,
- *hálózati topológia:* a mesterséges neuronok kapcsolati rendszere (egy neuron kimenete egy vagy több másik neuron egyik bemenetére kapcsolódhat),
- *tanulási szabály:* egy neuron számítási képletét módosító eljárás.

Szokásos topológiák.

- Előrecsatolt, rétegzett topológia.
- Visszacsatolt, rétegzett topológia.
- Hopfield-topológia.

Tanulási szabályok.

- Felügyelt (ismert a várt kimenet).
- Felügyelet nélküli (a várt kimenet nem ismert).
- Analaitikus.

Perceptron-modell

10.7.1. Definíció (Általánosított perceptron). *Belső memória nélküli mesterséges neuron, aktivizációs függvénye lehet pl. lépcsős vagy logisztikus, számítási képlete pedig skalárisszorzat-alakú ($\sum_{i=0}^{n} w_i x_i$).*

10.7.2. Definíció (Perceptron-modell). *Általánosított, lépcsős aktivizációs függvényű perceptronok egyrétegű, előrecsatolt hálója, felügyelt tanulással.*

Korlátok. A perceptron-modell csak *lineárisan szeparálható* problémák megoldására alkalmas.

Backpropagation-modell

10.7.3. Definíció (Backpropagation-modell). *Logisztikus aktivizációs függvényű, belső állapot nélküli neuronok több rétegű előrecsatolt hálója, speciális felügyelt tanulással.*

10.8. Automatikus logikai következtetés

10.8.1. Rezolúció

A logikai fogalmak formális bevezetése és a rezolúció módszerének tárgyalása a *Logikáról* szóló, 19. fejezetben található.

Kapcsolódó MI fogalmak. Egyesítő helyettesítés, legáltalánosabb egyesítő, egyesítő algoritmus.

Rezolúciós stratégiák

szélességi stratégia	rezolvensek sorfolytonos előállítása	sorrendi stratégia
klózok hossza szerint	a rezolválható klózok közül mindig a legkisebbet rezolváljuk	sorrendi stratégia
egységklóz-stratégia	a rezolválandó klózpár egyik fele mindig legyen egységklóz	csak Horn-klózok esetén teljes
lineáris input stratégia	az egyik szülő az előzőleg rezolvált klóz, a másik pedig a kiinduló halmazból származik	csak Horn-klózok esetén teljes
ősre korlátozott stratégia	a rezolvens klóz egyik szülője a kiinduló klózhalmazból való, vagy a másik szülő egyik őse	teljes
támogató halmaz stratégia	a rezolvált klózpár egyike a kiinduló halmaz egy kiválasztott (támogató) részhalmazából származik (sokszor a cél negáltjából származó klózok)	teljes, ha a támogató halmazon kívül kiinduló klózhalmaz kielégíthető

Hatékonyság javítása

érvényes klózok elhagyása	*érvényes klóz:* minden interpretációban igaz
befoglaló klózok elhagyása	egy klóz *bennfoglal* egy másikat, ha létezik helyettesítés, amelyet a befoglalt klózra alkalmazva az részformulája a befoglalónak
idegen literált tartalmazó klózok elhagyása	*idegen literál:* olyan literál, mely egysetlen másik klóz komplemens literáljával sem rezolválható

Válaszadás rezolúcióval

Alapelv. A célállítás klózait negáltjukkal egészítjük ki. Ezek a negált klózok nem vesznek részt a rezolúcióban, de a megfelelő helyettesítéseket végre-

hajtjuk rajtuk. Az eljárás végén az üres klóz helyett a „válaszformula” jelenik meg.

10.8.2. Alternatív következtetési formák

Az alábbi módszerek a rezolúció „szigorát” igyekszenek csökkenteni bizonytalan helyzetekben (pl. nem teljes vagy túl szigorú axiómarendszer).

Nemmonoton következtetés

Fogalmak.

- Ami nem bizonyítható, az nem igaz.
- Kivételkezelés (default logikai szabály) $\alpha(x) : \beta(x) \to \gamma(x)$

Valószínűségi következtetés

Fogalmak.

- Valószínűségi háló (feltételes valószínűség alapján).

Procedurális tudásábrázolás

Fogalmak.

- Deklaratív helyett procedurális szabályok (ha *feltétel*, akkor *következmény*) – az átírás nem nehéz.

Objektumalapú következtetés

Fogalmak.

- Objektumok és osztályok, *az-egy* kapcsolat.
- Első-, másod- és harmadrendű attribútumok (tulajdonságok).
- Öröklődés.
- Prioritások (többszörös származtatás esetén).
- Tudásbázis és a kérdés illesztése (legrövidebb út az öröklődések mentén, a prioritások figyelembe vételével).

Értékelés. Kifejező ereje az elsőrendű logikától elmarad, de a következtetési módszer jobban illeszkedik az emberi gondolkodáshoz.