

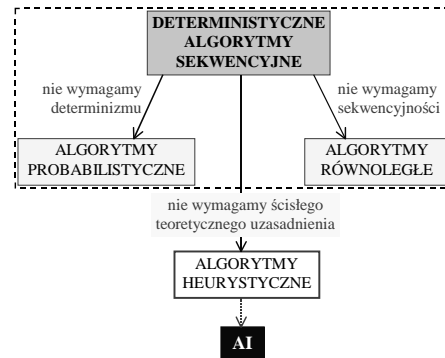
Algorytmiczna inteligencja

czyli tzw. sztuczna inteligencja (ang. AI)

Przedmiotem rozważań są relacje pomiędzy obliczeniami maszynowymi a ludzką inteligencją w najrozmaitszych kontekstach:

- Naśladowania
- Wspierania
- Zastępowania
- Analizowania
- Uczenia się
- Gromadzenia wiedzy
- Wnioskowania itd.

W obszarze informatyki zadanie AI jest często utożsamiane z „budowaniem maszyn, o których działaniu dałoby się powiedzieć, że jest podobne do ludzkich przejawów inteligencji”



Algorytmy heurystyczne są bardzo przydatne, bo odzwierciedlają szczególną naturę sterowania w inteligentnych programach: brak pewności, nieprecyzyjne dane, łatwość modyfikacji, konieczność adaptacji, możliwość ewolucji itp.

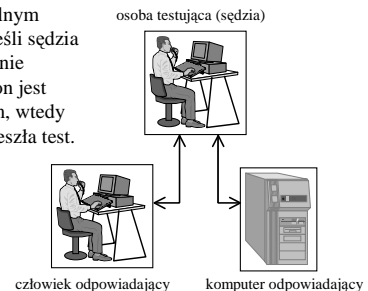
Dane, na których pracują algorytmy inteligentne muszą umożliwiać reprezentację i manipulowanie **wiedzą** w jej najrozmaitszych postaciach.

Osiągnięcie oryginalności rozwiązania charakterystycznej dla inteligentnego ludzkiego myślenia jest trudne do pogodzenia z zaprogramowanym z góry działaniem komputera.

Wprawdzie w szachach programy komputerowe osiągnęły poziom mistrzowski lub nawet arcymistrzowski, ale w *go* grają wciąż kiepsko, a i w brydża sportowego nie lepiej.

Test Turinga

Człowiek (sędzia) prowadzi rozmowę w języku naturalnym z pozostałymi stronami. Jeśli sędzia nie jest w stanie wiarygodnie określić, czy któraś ze stron jest maszyną, czy człowiekiem, wtedy mówi się, że maszyna przeszła test. Zakłada się, że zarówno człowiek jak i maszyna próbują przejść test jako człowiek.



Test zaproponował Alan Turing w 1950 r.

Przewidywał, że maszyny w końcu będą w stanie przejść ten test. Oceniał, że około roku 2000 maszyny będą w stanie oszukać 30% ludzkich sędziów w czasie pięciominutowego testu. Przepowiedział również, że ludzie przestaną uważać zdanie "myśląca maszyna" za wewnętrznie sprzeczne. Przewidywał także, że uczenie maszynowe nabierze dużego znaczenia w budowaniu wydajnych maszyn. To twierdzenie jest dzisiaj oceniane jako zasadne.

Uczenie maszynowe (ang. machine learning) jest dziedziną AI, w której przedmiotem zainteresowania są metody umożliwiające uczenie się programom komputerowym. Program uczy się, jeśli potrafi modyfikować jakiś aspekt samego siebie i dzięki temu stopniowo działa coraz lepiej lub wydajniej.

Przykładowe pytania zadawane przez sędziego:

- Czy jesteś komputerem?
- Która jest godzina?
- Kiedy była bitwa pod Grunwaldem?
- Ile jest 57585063836034546 razy 85756459?
- Czy w podanym problemie szachowym białe mogą dać matę w trzech posunięciach?
- Opisz swoich rodziców?
- Jakie wrażenie wywołują na tobie utwory The Doors?
- Co sądzisz o aktualnych wydarzeniach politycznych?
- Jaka jest twoja opinia o procesie globalizacji?

Najtrudniejsze są trzy ostatnie pytania!

Przykład „podróbki” ludzkiej inteligencji
(program ELIZA i typowa dla niego wymiana zdań)

Człowiek: Moja żona skarży się, że często jestem **przygnębiony**.
ELIZA: Przykro mi słyszeć, że jesteś przygnębiony.
Człowiek: Prawdę mówiąc, czuję się fatalnie – jestem **nieszczęśliwy**.
ELIZA: Czy myślisz, że po wizycie tutaj przestaniesz być nieszczęśliwy?
Człowiek: Być może pomożesz mi odkryć, dlaczego nienawidzę swojego **brata**.
ELIZA: Opowiedz mi coś więcej o swojej rodzinie.

Przy pozytywnym nastawieniu rozmawiającego program ELIZA może przez chwilę uchodzić za inteligentny, ale dość łatwo się gubi:

Człowiek: Byłem **bratem** zakonnym w klasztorze Benedyktynów.
ELIZA: Opowiedz mi coś więcej o swojej rodzinie.

Aby przeprowadzić rzeczywiście inteligentną rozmowę maszyna-kandydat musi mieć odpowiednio duży zasób **wiedzy**, posiadać zdolność **uczenia się** całkowicie nowych pojęć, konfrontowania ich z tym, co już wie i wyciągania trafnych wniosków.

Obecnie najlepszy w teście Turinga jest stale rozwijany program ALICE – stara się on w pełni naśladować heurystykę zwykłej konwersacji.

Na przykład:

Testujący: Co to jest krzesławka?
Maszyna: Nie mam pojęcia.
Testujący: Jest to gatunek afrykańskiej mrówki, którego osobniki potrafią szczipając się ze sobą imitować krzesło. Zastawiają w ten sposób pułapkę na zmęczonych podróżników. Kiedy któryś skusi się na odpoczynek, krzesławki opadają go, paraliżują ukąszeniami i pożerają.
Maszyna: Niesamowite!
Testujący: Czy sądzisz, że krzesławka występuje w przyrodzie?
Maszyna: Nie, to jest niemożliwe.
Testujący: Dlaczego?
Maszyna: Z wielu powodów. Po pierwsze, mrówki nie mogą mieć pojęcia co to jest krzesło i do czego służy. Po drugie, jest niemożliwe, aby potrafiły utrzymać formę krzesła normalnych rozmiarów. Po trzecie, nikt przy zdrowych zmysłach nie usiadłby na czymś takim. Ale to jest śmieszne!

Badania w zakresie AI prowadzone są zasadniczo w dwóch obszarach:

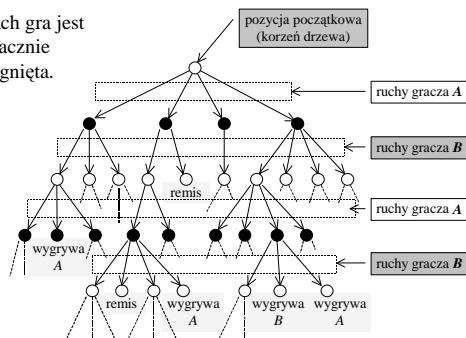
- wyjaśnianie w jaki sposób ludzie uczą się, rozumują, wnioskuje, planują, decydują itp.
- tworzenie programów komputerowych, które w pewnych zawężonych (na razie?) dziedzinach zachowują się inteligentnie.

Spore sukcesy osiągnięto w rozgrywaniu niektórych gier:

- znakomite programy grające w warcaby,
- bardzo dobre programy grające w szachy.

Drzewo gry (czyli sposób na zapisanie rozgrywki)

W liściach gra jest jednoznacznie rozstrzygnięta.



Drzewo gry w „kółko i krzyżyk”:
(dla tablicy 3x3)

- maksymalna głębokość – 9
 - maksymalna liczba potomków – 9 (w korzeniu, potem maleje)
 - maks. liczba wierzchołków do sprawdzenia $9! = 362\,880$
- ⇒ można napisać program, który gra perfekcyjnie

Drzewo gry w szachy:

- 20 możliwych otwarć białych
 - średnia liczba potomków – 35
 - głębokość drzewa rzędu 100
 - liczba wierzchołków do sprawdzenia w typowej grze – 35^{100}
- ⇒ nie da się napisać programu, który przeszukałby całe drzewo gry; trzeba stosować **heurystyki**

Co oznacza postępowanie heurystyczne?

Jeśli upuścimy na podłogę nieduży przedmiot i stracimy go z oczu, to możemy poszukiwać go różnymi metodami:

- systematycznie - deterministycznym algorytmem sekwencyjnym,
- „na chybił-trafił” - algorytmem probabilistycznym,
- systematycznie z pomocą przyjaciół - algorytmem równoległym,
- analitycznie - algorytmem rozwiązującym równania ruchu przedmiotu (pełen model fizyczny),
- intuicyjnie - **heurystycznym** algorytmem przeszukiwania zawężonego obszaru, w którym przedmiot może się znajdować (obszaru wyznaczonego np. na podstawie wcześniejszych doświadczeń)

Zalety algorytmów heurystycznych:

- możliwość radykalnego skrócenia czasu wykonania algorytmu,
- łatwość poprawiania i doskonalenia w miarę zdobywania wiedzy o problemie.

Wady algorytmów heurystycznych:

- nie gwarantują sukcesu,
- ich działanie nie daje się teoretycznie analizować – trzeba wypróbować i sprawdzać
- nie można oszacować teoretycznego prawdopodobieństwa sukcesu (co jest możliwe w algorytmach probabilistycznych)

Wiele heurystyk opiera się wprawdzie na ścisłych modelach i matematycznych sformułowaniach, ale nie dla całości problemu i dlatego nie dają one gwarancji sukcesu, np.:

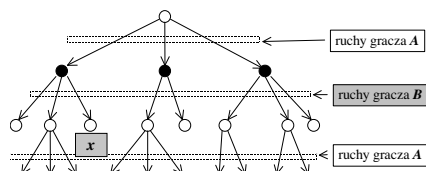
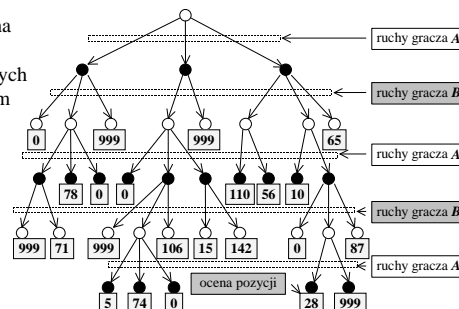
- komputerowe rozpoznawanie obrazu i mowy,
- sterowanie ruchem robotów i manipulatorów.

Wiele algorytmów optymalizacyjnych to w istocie rzeczy heurystyki, które można nazwać przypuszczalnymi algorytmami przybliżonymi, jak np.:

- rozwiązywanie bardzo dużych zadań transportowych,
- projektowanie układów VLSI,
- rozwiązywanie złożonych zadań harmonogramowania.

Przeszukując drzewo gry w celu znalezienia najlepszego posunięcia często stosuje się **strategię minimaksową**

Opiera się ona na ocenach przypisywanych wierzchołkom drzewa gry



x = ocena szans wygrania gry przez A,
 $x = 999$ – wygrana A,
 $x = 0$ – wygrana B.

Zasady strategii minimaksowej:

- A wybiera zawsze ruch maksymalizujący minimalną ocenę następnych pozycji (wierzchołków),
- B wybiera zawsze ruch minimalizujący maksymalną ocenę następnych pozycji (wierzchołków).

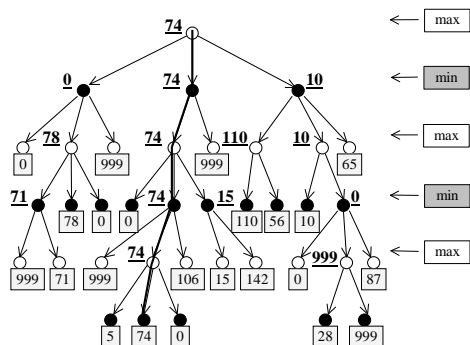
W analizie drzewa gry przy strategii minimaksowej następuje propagacja ocen wierzchołków w stronę korzenia:

- jako ocenę wierzchołka reprezentującego ruch gracza A przyjmuje się **maksimum** z ocen wierzchołków potomnych,
- jako ocenę wierzchołka reprezentującego ruch gracza B przyjmuje się **minimum** z ocen wierzchołków potomnych.

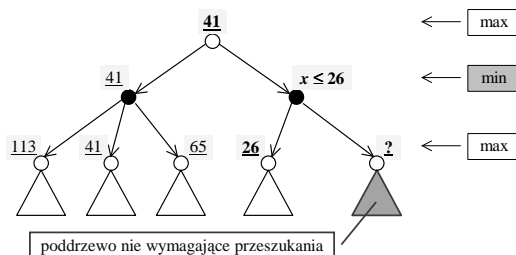
Wprawdzie w liściach drzewa gra jest jednoznacznie rozstrzygnięta, ale ogromna liczba liści np. w szachach ($\sim 35^{100}$) uniemożliwia rozpoczęcie od nich propagacji ocen.

Gdyby było to możliwe, to można by napisać program perfekcyjnie grający w szachy.

Przykład propagacji ocen w drzewie gry



Nie zawsze trzeba przeszukiwać wszystkie poddrzewa w celu wyznaczenia oceny wierzchołka:



Heurystyczne metody przeszukiwania drzewa gry zawierają:

- heurystyczne algorytmy wyznaczania ocen wierzchołków (stosuje się np. funkcje oceny o wielu parametrach, których wartości dobiera się heurystycznie i często mogą być one różne na różnych poziomach drzewa),
- heurystyczne reguły określające jak głęboko od pozycji bieżącej będzie sięgało przeglądanie poddrzew,
- efektywne procedury propagacji ocen, które pozwalają wykluczyć wiele poddrzew i zapewniają wyznaczenie w rozsądnym czasie wartości stanowiącej podstawę wyboru kolejnego ruchu.

Problemy dotyczące reprezentacji wiedzy w inteligentnych programach:

- dane potrzebne dla inteligentnych programów składają się z wielkich zbiorów faktów i złożonych związków pomiędzy nimi,
- związki pomiędzy faktami mają różne aspekty i własności, które tworzą sieci powiązań wyższego rzędu niż sama struktura danych,
- wciąż niewiele wiadomo o tym, w jaki sposób umysł ludzki magazynuje i wykorzystuje wiedzę nabywaną w ciągu całego życia,
- zaproponowano wiele modeli wiedzy, ale wszystkie one mają ograniczony zakres stosowania i nie ma wśród nich żadnego prawdziwie uniwersalnego i w pełni operacyjnego,
- opracowano języki programowania zawierające operacje pozwalające na manipulowanie wiedzą np. Lisp i Prolog.

Systemy ekspertowe (doradcze) są specyficzną odmianą inteligentnych programów przeznaczoną zwykle do współpracy z człowiekiem-ekspertem lub decydem, rzadziej do jego zastąpienia.

Znajdują zastosowanie np. w:

- diagnostyce medycznej,
- wojskowych systemach dowodzenia (podobno)
- w złożonych systemach zarządzania przedsiębiorstwami.

Podstawowe moduły składające się na system ekspertowy to:

- baza danych (zbiór faktów),
- baza wiedzy (zbiór reguł wnioskowania),
- maszyna wnioskująca (sterowanie i manipulowanie wiedzą).

Programy uczące się (uczenie maszynowe) są próbą naśladowania procesów zdobywania wiedzy przez ludzi.

Uczenie maszynowe znajduje zastosowanie:

- przy rozwiązywaniu problemów, dla których ze względu na ich złożoność niemożliwe jest podanie gotowego programu rozwiązującego dany problem w rozsądnym czasie,
- przy poszukiwaniu zależności w dużych zbiorach nieprzetworzonych danych (tzw. drążenie danych),
- przy tworzeniu inteligentnych autonomicznych systemów, umiających dostosowywać się do zmian w środowisku (roboty).

Kluczowym zagadnieniem w uczeniu maszynowym jest reprezentacja zdobywanej wiedzy.

Rozumienie języka naturalnego jest jak dotąd zagadnieniem bardzo trudnym do rozwiązania przy budowie inteligentnego systemu komputerowego.

Na maszynowe rozumienie języka naturalnego składa się m.in. innymi:

- rozpoznawanie mowy ludzkiej (np. wydzielenie słów),
- interpretacja semantyczna (znaczeniowa) wypowiedzi,
- uchwycenie związków międzywyrazowych (kontekstu).

Rozpoznawanie mowy:

umyj głowę i ręce ≈ umyj głowę Irence
Piotr chodzi z Anią ≈ Piotr chodzi za nią
mama ma nastroje ≈ mama ma nas troje

Jakie słowa zostały wypowiedziane?

Interpretacja semantyczna:

„usiadł przy stole i zobaczył przed sobą wielką porcję sałatki na talerzu obok kosza z pieczywem. Zajęło mu to trochę czasu, ale w końcu zdołał zjeść wszystko.”

Co „on” zjadł?

„Jan wyrzuci śmieci za karę”

„Jan wyrzuci śmieci za chałupę”

„Jan wyrzuci śmieci za godzinę”

Jak rozróżnić przyczynę, miejsce i czas?

Związki międzywyrazowe:

„dzieci ukradły monety, po czym niektóre z nich sprzedano”

„dzieci ukradły monety, po czym niektóre z nich złapano”

„dzieci ukradły monety, po czym niektóre z nich znaleziono”

Które wyrazy tworzą związek?