

Obsah

Přístupy umělé inteligence	3
Popis jednotlivých přístupů UI, základní charakteristika.....	3
Možnosti nasazení UI, příklady použití.....	3
Srovnání s klasickými přístupy řešení problémů vs. UI	3
Historický vývoj.....	3
Úvod do expertních systémů (ES).....	4
Vysvětlení pojmu expertní systém, charakteristické rysy ES	4
Schéma struktury expertního systému	4
Charakteristika základních složek expertního systému.....	5
Typy expertních systémů.....	5
Podmínky vhodnosti nasazení ES, výhody a nevýhody ES	5
Reprezentace znalostí	5
Reprezentace znalostí pomocí pravidel	5
Sémantická síť, vztahy v sémantické síti	5
Rámce, struktura rámce, vztahy mezi rámci	6
Reprezentace znalostí pomocí objektů	6
Srovnání způsobů reprezentace znalostí.....	6
Inferenční mechanismy	6
Základní strategie a struktury používané při usuzování v PS	6
Základní kroky dopředného řetězení v pravidlových systémech.	6
Algoritmus Rete.....	7
Algoritmus zpětného řetězení v pravidlových systémech.....	7
Zpracování neurčitosti.....	7
Srovnání přístupů ke zpracování neurčitosti v ES	7
Použití fuzzy logiky při zpracování neurčitosti	7
Bayesovský přístup ke zpracování neurčitosti a prosectorovské systémy.....	7
Přístup ke zpracování neurčitosti založený na faktorech jistoty	8
Dempster-Shaferova teorie.....	8
Bayesovské sítě.	8
Tvorba expertních systémů	8
Životní cyklus expertního systému a srovnání se softwarovým inženýrstvím.	8
Charakteristika prvních čtyř etap životního cyklu ES.....	8
Nástroje pro tvorbu expertních systémů a kritéria pro jejich výběr.....	9
Proces získávání znalostí a používané způsoby získávání znalostí.	9
Přehled a charakteristika technik strojového učení.....	9
Neuronové sítě	9
Rozdělení neuronových sítí, popis přístupu. Charakteristika jednotlivých NS, specifika	9
Popis algoritmu BackPropagation (Trénování vícevrstvého perceptronu)	9
Učení NS (s učitelem, bez učitele), chybová funkce, vybavování.....	10
Možnosti nasazení neuronových sítí, zhodnocení.	10
Evoluční a genetické algoritmy	10
Genetické algoritmy, popis algoritmu a možnosti nasazení.	10
Populace, jedinec, chromozom, mutace.....	10
Optimalizace funkcí, další možnosti nasazení.	11
Teorie her a prohledávání stavového prostoru.....	11

Popis jednotlivých přístupů a možnosti nasazení.	11
Rozdělení metod, klady a omezující kritéria.	11
Data mining	11
Co se rozumí pod pojmem dolování z dat, co tento obor obsahuje a čím se zabývá?	11
Uvedte netriviální příklad toho, kdy velmi různé datové struktury poskytují identické výsledky i z hlediska statistiky druhého řádu (střed, kovariance).	12
Co je to rozhodovací strom, jakou datovou strukturu představuje, k čemu se používá? Z jakých prvků se skládá – vysvětlete funkci těchto prvků.	12
Čím se zabývá induktivní strojové učení, na čem je založeno, jak a co se stroj učí?.....	12
Co je to entropie, jak je definována, k čemu, jak a proč se používá ve strojovém učení?	12
Vysvětlete, jak a proč se algoritmy strojového učení trénují a testují. Které testovací metody znáte a kdy a proč se používají, v čem se liší, jaké mají výhody a nevýhody?.....	12
Prolog	13
Vysvětlete pojem konstanta, proměnná vázaná, proměnná volná, příklady.	13
Predikáty, ukládání znalosti, definice pravidla.	13
Co je to klauzule, možnosti zápisu.	13
Operátory, definice.	13
Struktura, funktor a arita, vysvětlete – příklad.	13
Použití seznamů, rekurzivní přístup při psaní v rámci logického programování.....	13
Popište postup prologu při vyhodnocování a plnění cílů.	13
Popište operaci srovnání (unifikace) v Prologu.	13
Popište blokový model vyhodnocování v Prologu.	13
MATLAB	13
Popis Matlabu a jeho využití.	13
Definice skriptů za pomoci M souborů, tvorba podprogramů.....	13
Tvorba proměnných, řádkových a sloupcových vektorů, matic.....	14
Vykreslování 2D/3D grafů a optimalizace znázornění.....	14
Cykly, podmínky, chráněné bloky.....	15
Otázkové okruhy bez zařazení	16
Programovací techniky pro umělou inteligenci: Sémantiky programovacích jazyků.....	16
Logické programování a jazyk Prolog: řízení výpočtu, řez, seznamy, vestavěné predikáty, styl programování, optimalizační techniky.	16
Metody reprezentace znalostí a inference: reprezentace znalostí, pravidla, rámce, sémantické sítě.....	16
Metody tvorby báze znalostí.	16
Prolog a reprezentace znalostí.	16
Hry a základní herní strategie.....	16
Prohledávání a programování s omezujícími podmínkami: řešení problémů využitím vyhledávacích metod, prohledávání stavového prostoru.....	16
Stromové prohledávání, lokální prohledávání, neúplné prohledávání.....	16
Řešení optimalizačních problémů.	16
Dolování z dat a strojové učení: proces dobývání znalostí z dat a dolování z dat. Typické úlohy.	16
Předzpracování dat. Strojové učení.....	16
Neuronové sítě a genetické algoritmy.	16
Posilované učení (reinforcement learning).	16
Vizualizace dat.....	16
Metody strojového učení pro zpracování textu.....	16
Klasifikace a filtrace dokumentů. Extrakce informace z textu.	16

Přístupy umělé inteligence

Popis jednotlivých přístupů UI, základní charakteristika

Vytvořit obecnou UI je velmi obtížné, vědci ovšem již vyvinuli řadu postupů, které dosahují dílčích úspěchů v jednotlivých problémech.

Neuronové sítě – Skládají se z výpočetních modelů neuronů, které si navzájem předávají signály a transformují je pomocí funkce pro přenos k dalším „neuronům“.

Genetické programování – Postup, jenž namísto sepsání konkrétního algoritmu pro řešení úkolu hledá tento postup evolučními metodami.

Expertní systémy – Počítačový program, který má za úkol poskytovat expertní rady, rozhodnutí nebo doporučit řešení v konkrétní situaci.

Prohledávání stavového prostoru – Hledáme cestu od počátečních stavů ke koncovým stavům, které znamenají náš úspěch.

Dobývání znalostí – Převádí velké soubory dat o nějakém systému do kompaktní a explicitní formy popisující systém, která je lépe použitelná. Součástí je obvykle strojové učení.

Strojové učení – Umožňuje počítačovému systému učit se. To je taková změna vnitřního stavu systému, která zefektivní schopnost přizpůsobení se změnám okolního prostředí.

Možnosti nasazení UI, příklady použití

Expertní systémy – lékařství, firemní znalosti atd.

Prohledávání stavového prostoru – klasické deskové hry – šachy, dáma, piškvorky, go (omezeně)

Robotika a další aplikační použití

- Řešení úloh v reálném čase
- Plánování v reálném čase
- Počítačové vidění
- Automatické porozumění přirozenému jazyku
- Systémy virtuální reality

Řešení optimalizačních problémů

- Nalezení nejkratší cesty (problém obchodního cestujícího, zásobování)
- Nalezení optimální cesty (plánování cesty výrobku při výrobě)
- Optimalizace dráhy nástroje
- Optimalizace prostoru (výběr přepravní nádoby dle výrobku)
- Optimalizace nelineárních soustav (regulátory)

Identifikace systémů

- Rozeznávání obrazů a počítačové vidění (hlídací kamerové systémy, kooperující roboti, ...) - NS

Predikce

- Časových řad (finanční ukazatele, měnové kurzy, ...) - NS

Srovnání s klasickými přístupy řešení problémů vs. UI

UI se používá tam, kde je klasický přístup příliš pomalý nebo není vůbec možný.

Příklad – piškvorky

Klasické řešení hrubou silou (pomocí podmínek IF) – délka programu a časová složitost roste dle velikosti herní plochy (faktoriálem). Volbou vhodné strategie lze složitost snížit.

Lepší řešení – transformace úlohy na stavový prostor. Definujeme stavy problému, možné přechody mezi stavy, cílové a počáteční stavy. Reprezentace pomocí stromového grafu.

Historický vývoj

Otázkou, zda mohou stroje myslet se již v 17. století zabývali významní filozofové jako Pascal, Hobbes a Descartes. Neřešeno, jak toho dosáhnout.

1956 – programy používající heuristické prohledávání (Carnegie Institute of Technology)

Následně ambiciózní předpovědi pro rok 1970, které později vyvolaly tzv. krizi UI.

NS – konec 50. let – Rosenblatt vyvinul perceptron.

V 70. letech kritika – obnova později Hopfield, Kohonen.

Jazyky – LISP (50. Léta), Prolog – 1967

70. léta – expertní systémy: prototypy MYCIN a PROSPECTOR, komerční od 1981

Japonský projekt a nová generace expertních systémů

Úvod do expertních systémů (ES)

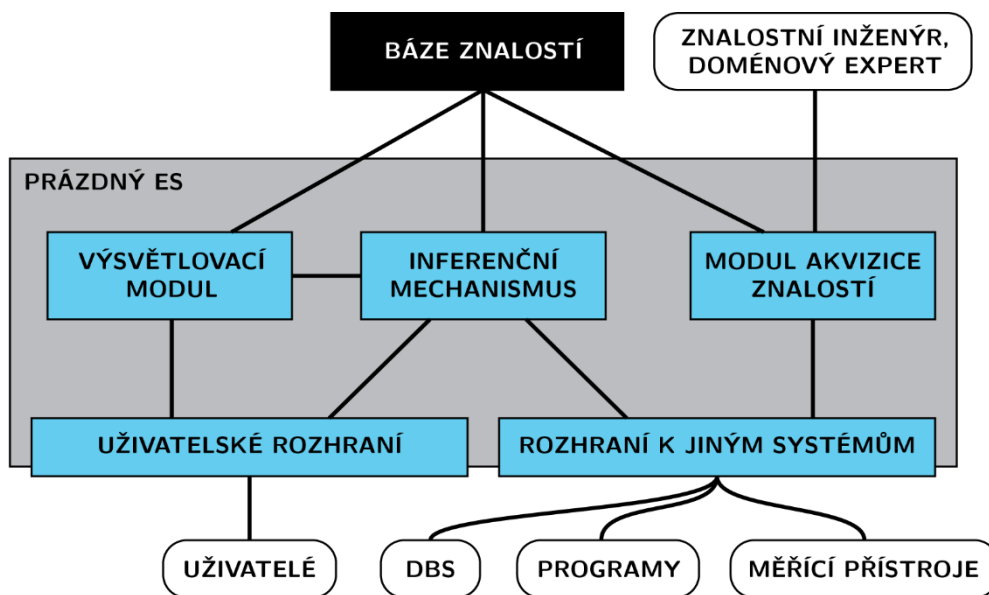
Vysvětlení pojmu expertní systém, charakteristické rysy ES

Expertní systémy jsou počítačové programy vystavěné s cílem **dosáhnout stejné rozhodovací úrovně jako lidský expert v dané oblasti** (doméně). Využívají znalosti získané od experta.

Rysy ES:

- oddělení znalostí a mechanismu jejich využívání
- rozhodování za stavu neurčitosti
- schopnost vysvětlování

Schéma struktury expertního systému



Charakteristika základních složek expertního systému

Základní složky ES

- **báze znalostí**, znalosti z určité domény a specifické znalosti o řešení problémů v této doméně
- **inferenční mechanismus**, obsahuje obecné (doménově nezávislé) algoritmy schopné řešit problémy na základě manipulace se znalostmi z báze znalostí.
- **I/O rozhraní** (uživatelské, vývojové, vazby na jiné systémy),
- **vysvětlovací modul**,
- **modul pro akvizici znalostí**, modul pro získání a reprezentace znalostí

Báze faktů – Vytváří se v průběhu řešení konkrétního problému a obsahuje data k problému.

Typy expertních systémů.

Problémově orientovaný ES: báze znalostí obsahuje znalosti z určité domény.

Prázdný ES (Shell): báze znalostí je prázdná.

Diagnostický ES: jeho úkolem je určit, která z hypotéza z předem definované konečné množiny cílových hypotéz nejlépe koresponduje s daty týkajícími se daného konkrétního případu.

Plánovací ES: obvykle řeší takové úlohy, kdy je znám cíl řešení a počáteční stav a je třeba s využitím dat o konkrétním řešeném případě nalézt posloupnost kroků, kterými lze cíle dosáhnout.

Podmínky vhodnosti nasazení ES, výhody a nevýhody ES

Podmínky

Musí se jednat o problém složitý rozsahem nebo neurčitostí vztahů, pro nějž exaktní metoda řešení buď není k dispozici, nebo není schopna poskytnout řešení v požadované době.

Efekty plynoucí z použití expertního systému musejí převyšovat vynaložené náklady. To znamená, že by mělo jít o problém s opakovanou potřebou řešení a značnými finančními dopady, pro nějž lidští experti jsou drazí nebo omezeně dostupní.

Výhody

schopnost řešit složité problémy,
dostupnost expertíz a snížené náklady na jejich provedení,
trvalost a opakovatelnost expertízy,
stabilita výsledků,
uchování znalostí odborníků odcházejících z organizace.

Nevýhody

nebezpečí selhání ve změněných podmínkách,
neschopnost poznat meze své použitelnosti,
již nachystaný systém je vhodný pouze pro konkrétní oblast,
nejsou schopní získávat nové informace, někdo je musí učit,
nutnost aktualizaci softwaru.

Reprezentace znalostí

Reprezentace znalostí pomocí pravidel

Tvar pravidla: IF předpoklad THEN závěr ... E → H (evidence → hypotéza)

levá strana – antecedent – podmínková část: více podmínek přes AND, OR

pravá strana – konsekvent – více důsledků (akce) přes AND

Příklad: IF auto_startuje = ne AND světla_svítily = ne THEN diagnóza = vybitá_baterie

může také obsahovat neurčitosti

Sémantická síť, vztahy v sémantické síti

SS je **ohodnocený orientovaný graf** – uzly představují objekty a hrany vztahy mezi objekty.

Poskytuje vyšší úroveň porozumění akcím, příčinám a událostem, které se vyskytují v doméně

Vztahy: je (is-a, ISA) – instance nebo podtřída (záleží na systému); je částí (part-of); má (has-a).

SS podporuje dědičnost (podtřída má všechny vlastnosti rodiče) a tranzitivitu.

Rámce, struktura rámce, vztahy mezi rámci

Rámce (frames) jsou struktury pro reprezentaci **stereotypních situací** a odpovídajících stereotypních činností (scénářů). Rámce se pokoušejí reprezentovat obecné znalosti o třídách objektů, znalosti pravidlé pro většinu případů. Mohou existovat objekty, které porušují některé vlastnosti popsané v obecném rámci.

Struktura rámce

Jméno a množina atributů (sloty) – ty mohou obsahovat různé položky: Aktuální hodnota, implicitní hodnota, rozsah možných hodnot

Procedury – aktivují se při určité akci: if-needed, if-changed, if-added, if-deleted

Vztahy

– **Rodič – potomek** (is-a, is-an, is-a-kind-of): Tento vztah může být typu 1:1, 1:n, n:1. Dědění některého atributu může být pro určitý rámec potlačeno.

– **Rámec – instance rámce** (is-an-instance-of). Tento vztah je typu 1:1. Přitom je navíc možné dědění nějakého specifického atributu od nějakého specifického rámce.

– **Vlastnictví rámce** Atributem rámce může být jiný rámec.

Reprezentace znalostí pomocí objektů

Objekt je programová struktura, obsahující jak data, tak metody (procedury), které s těmito daty pracují. Objekt je instance třídy (skupina objektů, které mají datové služby a metody).

Vztahy mezi třídami: dědičnost, skládání

Komunikace mezi objekty – posílání zpráv

Srovnání způsobů reprezentace znalostí

Pravidla – výhody: modularita, uniformita, přirozenost

Sémantická síť – výhody: explicitní a jasné vyjádření, redukce doby hledání

Rámce, objekty – nevýhody: potíže s odlišností objektů od prototypu, obtížné přizpůsobení novým situacím

Inferenční mechanismy

Obsahuje obecné (doménově nezávislé) algoritmy schopné řešit problémy na základě manipulace se znalostmi z báze znalostí.

Základní strategie a struktury používané při usuzování v PS

Řešení problému spočívá v **nalezení řady inferencí** (inference chain), které tvoří cestu od definice problému k jeho řešení.

Strategie

Usuzování řízené daty (dopředné řetězení) – při aplikaci produkčních pravidel postupujeme ve směru od počátečního stavu k některému ze stavů cílových (strategie řízená daty)

Usuzování řízené cíli (zpětné řetězení) – vychází se od cíle ve směru počátečních stavů (strategie řízená cílem)

Struktury

Inferenční síť – Závěry pravidel jsou fakta, která korespondují s předpoklady jiných pravidel. Může být prezentována jako graf, jehož uzly jsou fakta a orientované hrany pravidla.

Systém porovnávání se vzorem – Závěry pravidel jsou obecnější a můžeme je chápat jako kolekce faktů, které mohou nebo nemusí korespondovat se vzory popsanými v předpokladech jiných pravidel.

Základní kroky dopředného řetězení v pravidlových systémech.

V tomto algoritmu se opakují následující tři kroky:

1. **Porovnání** – Pravidla ze znalostní báze jsou porovnávána se známými fakty, aby se zjistilo, u kterých pravidel jsou splněny předpoklady.

2. **Řešení konfliktu** – Z množiny pravidel se splněnými předpoklady se vybírá pravidlo podle priority a v případě více pravidel se stejnou prioritou podle nějaké strategie:

Starší nebo novější fakta, složitější nebo jednodušší pravidla

3. **Provedení** – Provede se pravidlo vybrané v předchozím kroku.

Důsledek: např. přidání nebo odebrání faktu nebo pravidla

Algoritmus Rete.

Za použití algoritmu Rete expertní systém vytváří speciální graf nebo prefixní strom, kde uzly jsou představeny částmi podmínek pravidel. Cesta od kořenu k listu tvoří úplnou podmínku. V podstatě každý uzel obsahuje seznam faktů, odpovídající podmínce. Při přidání nebo modifikaci faktu se tahle změna prochází celým stromem, ve výsledku se označují listy, se kterými se shoduje daný fakt. Když je dosaženo listu, vykonává se celé pravidlo.

V porovnání s jinými algoritmy Rete zabírá více paměti, ale funguje efektivněji a rychleji.

Algoritmus zpětného řetězení v pravidlových systémech.

1. Utvoř **zásobník** a naplň jej všemi koncovými cíli.
2. Shromáždí všechna pravidla schopná splnit cíl na vrcholu zásobníku. Je-li zásobník prázdný, pak konec.
3. Zkoumej postupně všechna pravidla z předchozího kroku.

Možnosti:

- a) Všechny předpoklady splněny – proved' pravidlo, odstraň cíl a jdi na bod 2
- b) Fakta nesplňují předpoklady – zkoumání pravidla ukončeno
- c) Pro některý parametr předpokladu chybí hodnota v bázi faktů – najde se přes podcíl nebo zjistí od uživatele.
- d) Jestliže pomocí žádného ze zkoumaných pravidel nebylo možné odvodit hodnotu důsledku, pak daný cíl zůstává neurčen. Odstraní se ze zásobníku a pokračuje se krokem 2.

Zpracování neurčitosti

Neurčitost se může vyskytovat jednak v bázi znalostí a jednak v bázi faktů.

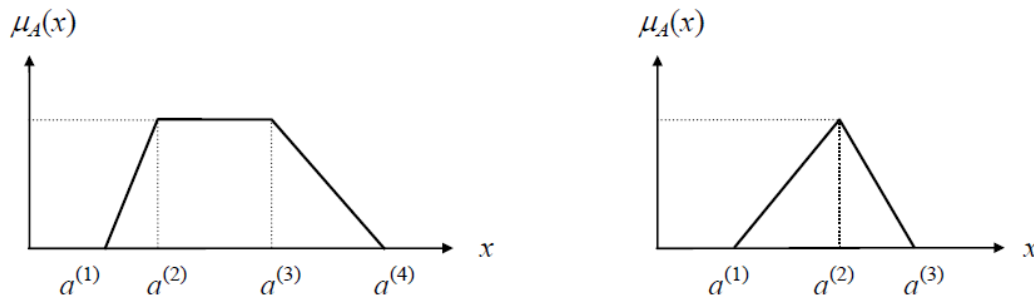
Srovnání přístupů ke zpracování neurčitosti v ES

přístupy založené na **ad-hoc modelech** – faktory jistoty, pseudo-bayesovské přístupy

přístupy založené na **teoretických principech** – teorie pravděpodobnosti, teorie fuzzy množin, teorie fuzzy míry (sem patří např. Dempster-Shaferova teorie a teorie možnosti)

Použití fuzzy logiky při zpracování neurčitosti

Fuzzy logika je vícehodnotová logika definovaná funkcí příslušnosti prvku na intervalu $\langle 0,1 \rangle$.



Fuzzifikace – Převádí reálné proměnné do jazykových proměnných.

Fuzzy inference – definování transformačních pravidel (př. IF cena = mala THEN velký prodej)

Defuzzifikace – Převod jazykových proměnných do reálných proměnných.

Použití v ES

Poskytování pojištění (zhodnocení rizik a navrhnutí adekvátního pojištění); fuzzy řízení výrobních procesů; podpora rozhodování investorů; predikce cenového pohybu dané komodity, popřípadě kurzu

Bayesovský přístup ke zpracování neurčitosti a prospectorovské systémy.

Neurčitost závěru H v závislosti na předpokladu E může být kvantifikována pomocí podmíněné pravděpodobnosti – $P(H|E)$.

Výhody a nevýhody bayesovských přístupů

- + dobré teoretické základy
- + dobře definovaná sémantika rozhodování
- potřeba velkého množství pravděpodobnostních dat

- nebezpečí neúplnosti a nekonzistence dat
- předpoklad nezávislosti evidencí E_i bývá v praxi zřídka splněn
- možnost ztráty informace v důsledku popisu neurčitosti jedním číslem
- obtížnost vysvětlování

Přístup ke zpracování neurčitosti založený na faktorech jistoty

Faktory jistoty (certainty factors) byly poprvé použity v systému MYCIN. Cílem bylo eliminovat některé slabiny čistě pravděpodobnostního přístupu.

Znalosti jsou vyjádřeny opět ve tvaru pravidel $E \rightarrow H$, přičemž s každým pravidlem je spojen faktor jistoty CF (je určen pomocí měr důvěry a nedůvěry MB a MD).

Faktor jistoty vyjadřuje **stupeň důvěry v hypotézu H**, jestliže předpoklad E je pravdivý (1 znamená absolutní důvěru, -1 absolutní nedůvěru).

Dempster-Shaferova teorie.

Pracuje s množinou podmnožin všech možných hypotéz. Umožňuje, aby nevědomost byla explicitně vyjádřena a udržována při aktualizacích. Subjektivní důvěra přiřazená události a její negaci musí být v součtu jedna.

Bayesovské sítě.

Je pravděpodobnostním modelem, který využívá grafovou reprezentaci pro zobrazení pravděpodobnostních vztahů mezi jednotlivými jevy.

Tvorba expertních systémů

Životní cyklus expertního systému a srovnání se softwarovým inženýrstvím.



Charakteristika prvních čtyř etap životního cyklu ES.

Analýza problému

Cílem analýzy je posoudit vhodnost aplikace znalostních technik pro řešení daného problému.

Specifikace požadavků

Zde dochází k celkovému vymezení expertního systému z hlediska řešeného problému, jeho cíle a funkcí, budoucích uživatelů, ale také omezení a dalších požadavků.

Předběžný návrh

Ten by měl rovněž vymezovat nástroj pro tvorbu zamýšleného systému. Předmětem předběžného návrhu je taky vymezení lidských zdrojů potřebných k realizaci ES.

Rychlé prototypování

Cílem rychlého prototypování je vytvořit v relativně krátké době fungující verzi systému, která poslouží k simulacím chodu systému a uživatelem a následnému posouzení aplikovatelnosti systému.

Nástroje pro tvorbu expertních systémů a kritéria pro jejich výběr.

Nástroje pro tvorbu expertních systémů

Prázdné expertní systémy: EXSYS, FLEX, G2, HUGIN, M4, ...

Speciální programová prostředí: CLIPS, OPS5, Lisp, Prolog, ...

Obecná programová prostředí: Pascal, Delphi, C, C++Builder, ...

Kritéria pro jejich výběr:

1. **Paradigma** reprezentace znalostí a usuzování.
2. **Flexibilita.** Uživatelsky definované funkce, externí rutiny, vestavěné funkce, podpora datových struktur.
3. **Speciální požadavky.** Časové usuzování, operace v reálném čase, zpracování neurčitosti, přístup k externímu softwaru, grafika, okna.
4. **Pomocné funkce.** Editor znalostní báze, trasování, vysvětlování, testovací a verifikační pomůcky, grafická prezentace znalostní báze.
5. **Výkon.**
6. **Podpora výrobce.** Dokumentace, on-line help, podpora horkou linkou, školení, konzultace.
7. **Náklady**

Proces získávání znalostí a používané způsoby získávání znalostí.

Získávání znalostí (knowledge acquisition) je klíčovou operací implementace ES a představuje nejdelší a nejpracnější část vývoje ES.

- Získávání znalostí od expertů formou spolupráce mezi znalostními inženýry a experty
- Automatizované získávání znalostí (strojové učení)
 - od expertů
 - z textů
 - z dat (data mining)

Přehled a charakteristika technik strojového učení.

učení s učitelem – pro vstupní data je určen správný výstup (třída pro klasifikaci nebo hodnota pro regresi)

učení bez učitele – ke vstupním datům není známý výstup

kombinace učení s učitelem a bez učitele – část vstupních dat je se známým výstupem, ale další data, typicky větší, jsou bez něj

Neuronové sítě

Rozdělení neuronových sítí, popis přístupu. Charakteristika jednotlivých NS, specifika.

Vícevrstvé NS – se skládá z vrstvy vstupní, skryté a výstupní, síť je tvořena opakováním perceptronu, počet vstupních neuronů je dán počtem vstupů matematického modelu, počet neuronů ve skryté vrstvě je volen s ohledem na složitost úlohy, obvykle jako maximum ze vstupů a výstupů, standardně je síť učena za pomoci učení s učitelem,

Samoorganizující se NS – jednovrstvá síť s dopředným šířením, učící algoritmus ve variantě "učení bez učitele" provádí pouze analýzu vstupních dat (shlukovou analýzu), NS obsahuje jedinou vrstvu radiálních neuronů, které jsou uspořádány do mřížky,

Hopfieldovy NS – výstup z každého neuronu je přiváděn na vstup všech ostatních neuronů (umožňuje postupně zpřesňovat výsledek), všechny neurony jsou jak vstupní, tak výstupní, míra vzájemné spojitosti neuronů je vyjádřena vahou, žádný neuron není spojen sám se sebou,

Radiální NS – je síť radiálních jednotek s pevným počtem vrstev, má rychlejší průběh učení, první vrstva se nepočítá mezi produkční vrstvy.

Popis algoritmu BackPropagation (Trénování vícevrstvého perceptronu)

1. Počáteční inicializace vah a prahů jednotlivých neuronů
2. Přivedení vstupního vektoru a definice požadované odezvy
3. Výpočet aktuálního výstupu
4. Adaptace vah a vztahů
5. Opakování kroků 3–5 dokud není chyba menší než předem stanovená hodnota

Učení NS (s učitelem, bez učitele), chybová funkce, vybavování.

Učení s učitelem

NS se snaží srovnáváním aktuálního výstupu s požadovaným výstupem (učitelem) přenastavit váhy sítě tak, aby se na daný konkrétní vstup snížil rozdíl mezi skutečným a požad. výstupem.

Učení bez učitele

Váhy sítě se nastavují tak, aby výstup byl konzistentní, tedy aby síť poskytovala stejnou odezvu na budící signál při stejných, nebo podobných vektorech vstupu.

Chybová funkce

Na začátku učení bývají váhy nejčastěji nastaveny na náhodná čísla generovaná v určitém intervalu, popřípadě je všem vahám přiřazena jedna hodnota.

Proces učení se snaží minimalizovat chybu sítě (rozdíl výstupních hodnot požadovaných od aktuálně získaných). Hodnota chybové funkce by měla u učící se funkce iteračně klesat k 0.

Možnosti nasazení neuronových sítí, zhodnocení.

predikce (vstupem jsou časové řady, využití trendové složky – predikce měnových kurzů, stavů soustavy),

klasifikace (výrobků, poskytnutí úvěru, rozhodování o poskytnutí a výši pojištění),

aproximace (funkcí – vychází ze základů neuronových sítí jako univerzálního aproximátoru).

Evoluční a genetické algoritmy

Genetické algoritmy, popis algoritmu a možnosti nasazení.

Genetický algoritmus (GA) je heuristický postup, vycházející z evolučního přístupu, který lze nasadit na řešení složitých problémů, pro které neexistuje použitelný exaktní algoritmus.

GA používají techniky napodobující evoluční procesy známé z biologie (přirozený výběr, křížení, dědičnost, mutace).

Použití – především optimalizační úlohy.

GA svým pojetím popisuje postupnou **tvorbu generací**, dílčích řešení daného problému (každý jedinec je jedno možné řešení). Při evoluci v populaci se řešení zlepšuje (při vhodném navržení algoritmu a tím spojenou konvergencí řešení).

Popis algoritmu

1. Vytvoření nulté generace – inicializace (obvykle náhodně vygenerována).

2. Selektce – volí se takoví jedinci, kteří jsou relevantní k dalšímu řešení.

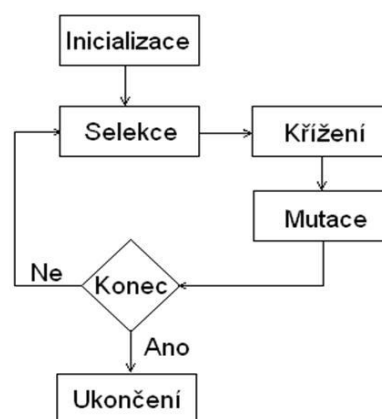
3. Generování nových jedinců – křížení, mutace.

4. Omezení počtu jedinců na velikost nastavené populace.

5. Vypočtení zdatnosti jedinců – fitness.

6. Ukončení cyklu – **konec**, pokud je splněna zastavovací podmínka (dosažení dané přesnosti nebo daný počet vytvořených populací), v opačném případě pokračuj bodem 2.

7. Při ukončení algoritmu se vybere jedinec s tzv. nejvyšší zdatností, ten pak reprezentuje nejlepší nalezené řešení.



Populace, jedinec, chromozom, mutace.

Jedinec = fenotyp, reprezentuje ho (to řešení) **chromozom**: ten se dělí na **geny**, možné hodnoty genu jsou **alely**

Znázornění chromozomu – **kódování** – nejčastěji binární: (1,0,1,0,1,1,0,0)

Nejčastěji aplikovaná selektce je tzv. turnajová, křížení binární, tj. mezi dvěma jedinci a mutace na úrovni cca 5 % obsahu zakódovaného v chromozomu jedince.

Optimalizace funkcí, další možnosti nasazení.

Optimalizace funkcí

GA se používá pro řešení problémů, u nichž výsledek není znám, mnohdy se jen tuší. Proto v průběhu výpočtu hrozí, že může algoritmus uvíznout v lokálním extrému (lokální minimum, lokální maximum) optimalizované funkce řešeného matematického modelu. Z tohoto důvodu se zavádí náhodná změna genomu jedince ve formě mutace. Snahou je naléztí globální hodnotu extrému, tj. řešení dané úlohy.

GA se používá u optimalizačních problémů, kde časová náročnost řešení nenarůstá lineárně s rostoucí složitostí (počtem prvků modelu), ale je zde nárůst exponenciální nebo faktoriální.

Ekonomické úlohy

Optimalizace výrobního plánu podniku; Výběr investic; Klasifikace podniku; Problém obchodního cestujícího; Úloha o baťohe; Umístění distribučního skladu

Teorie her a prohledávání stavového prostoru.

Popis jednotlivých přístupů a možnosti nasazení.

Použití

Úlohy s velkou výpočetní náročností, kde počet řešení roste se složitostí úlohy.

Algoritmy

prohledávání do hloubky, prohledávání do šířky, hledání prvního nejlepšího, A* algoritmy iterativního vylepšování: simulované žihání, gradientní sestup
nedeterministické algoritmy: neuronové sítě, genetické algoritmy
metody hraní her: mini-maxová procedura

Rozdělení metod, klady a omezující kritéria.

Prohledávání do šířky

Seznam OPEN a CLOSED, OPEN je fronta (FIFO).

Výhoda: Tímto postupem najdeme nejprve nejdříve dosažitelné cílové stavy (nejkratší cestu z výchozího do cílového stavu).

Nevýhoda: nutnost udržovat seznam uzlů OPEN – velká paměťová náročnost algoritmu

Prohledávání do hloubky

Seznam OPEN a CLOSED, OPEN je zásobník (LIFO).

Výhoda: Rychleji najde nějaké řešení (ne nutně nejlepší), je možné použít rekurzi.

Důležité v případě omezených zdrojů – lze ukončit dříve než prohl. do šířky.

Zvolíme cestu a v případě, že nenajdeme cílový stav, se vrátíme se do výchozího stavu (to se jmenuje backtracking) a začneme znovu jinou cestou.

Hledání prvního nejlepšího

Optimalizace – Znalosti o problému a strategie transformujeme na informovaný výběr stavů, které budou prohledány dříve.

Prohledávání do hloubky, ve kterém zvolíme „nadějnou“ cestu.

Jak? Co nejmenší odhadovaná cena dosažení cíle (cena cesty do cílového uzlu).

Heuristická funkce: $h(n)$ – odhadovaná cena nejlevnější cesty ze stavu v uzlu n do stavu v cíli

Data mining

Co se rozumí pod pojmem dolování z dat, co tento obor obsahuje a čím se zabývá?

Dolování z dat je zaměřeno na odkrytí znalosti v datech ukryté: data -> informace -> znalost.

Dolování z dat využívá veškeré vhodné technologie (postupy), zejména strojové učení, umělou inteligenci, logiku a matematiku. Používají se tam algoritmy strojového učení (Machine learning).

Zabývá se technologiemi prohledávání libovolných (reálných/abstraktních) prostorů; cílem je nalezení optima (globálního maxima), tedy nejlepšího řešení nějakého zadaného problému.

Uved'te netriviální příklad toho, kdy velmi různé datové struktury poskytují identické výsledky i z hlediska statistiky druhého řádu (střed, kovariance).



Data mají stejný střed + i kovarianci, tedy se jeví stejně. Ideálně by případy měly být zařazeny do příslušných shluků či kategorií odlišně.

Co je to rozhodovací strom, jakou datovou strukturu představuje, k čemu se používá? Z jakých prvků se skládá – vysvětlete funkci těchto prvků.

Data získaná libovolným způsobem nemají obecně nijak zřetelnou strukturu. Dolování z dat odhaluje různé skryté, avšak existující struktury — vzory struktur.

Rozhodovací stromy: hierarchický systém pravidel pro klasifikaci a vysvětlení variability cílové vlastnosti. Každý uzel stromu představuje rozhodování podle jedné (vybrané) vlastnosti objektu, z tohoto uzlu vede konečný počet hran. Každá větev zároveň představuje jedno pravidlo.

Čím se zabývá induktivní strojové učení, na čem je založeno, jak a co se stroj učí?

Na základě omezeného vzorku příkladů E+ a E-, charakterizovat (popsat) zamýšlenou skupinu objektů (koncept) tak, aby popis co nejlépe odpovídal právě prvkům z E+ byl použitelný pro určení i objekty mimo E.

Induktivní logické programování formuluje hypotézy pro zkoumané koncepty pomocí jazyka logiky prvního řádu, přesněji hypotézu tvoří konečná množina klauzul odpovídající programu v použitém systému logického programování (nejčastěji Prologu)

Co je to entropie, jak je definována, k čemu, jak a proč se používá ve strojovém učení?

Odpovědi na dotazy v testovacích uzlech stromu by měly dát co nejvíce informace, tj. odpovědi by měly být co nejužitečnější vzhledem k řešené úloze.

K dobrým a praktickým možnostem při hledání vhodných dotazů patří využití entropie. **Entropie** je míra neurčitosti systému.

Při tvorbě rozhodovacího stromu se hledají testy v uzlech tak, aby odpověď na test poskytla, co nejvíce informace, tj. aby byla co nejmenší entropie (mírou "překvapení")

Vysvětlete, jak a proč se algoritmy strojového učení trénují a testují. Které testovací metody znáte a kdy a proč se používají, v čem se liší, jaké mají výhody a nevýhody?

Pro odhad chyby klasifikátoru na budoucích datech, která nebyla k dispozici během trénování, je nutno klasifikátor otestovat.

Krosvalidace (cross-validation)

Krosvalidace se s úspěchem používá **na ověření datové stability vystavěného modelu.**

Metoda nejbližších sousedů k-NN

Blížkost je zde ekvivalentní podobnosti, takže se vlastně hledá k nejpodobnějším případům.

Bayesovské učení

Bayesova metoda inference je založena na použití teorie pravděpodobnosti. Strojové učení poskytuje kvantitativní přístup ke zvažování důkazů podporujících alternativní hypotézy.

Prolog

Vysvětlete pojem konstanta, proměnná vázaná, proměnná volná, příklady.

Konstanta – čísla (celá nebo reálná),

proměnná vázaná – nastavená (instanciovaná) proměnná má již přiřazenu konkrétní hodnotu,

proměnná volná – nenastavená (neinstanciovaná) proměnná není vázána na žádnou konstantu nebo strukturu jako na svoji hodnotu.

Predikáty, ukládání znalosti, definice pravidla.

Predikáty (fakta) – popisují vztahy mezi objekty či jejich vlastnostmi

ukládání znalosti:

definice pravidla: partner(Jmeno) :- manzele(Jmeno,_).

Co je to klauzule, možnosti zápisu.

Program v Prologu sestává z klauzulí. Klauzule definující stejnou relaci vytvářejí tzv. proceduru. Rozlišujeme následující typy klauzulí:

- Fakta: manzele(josef,marketa)
- Pravidla: partner(Jmeno) :- manzele(Jmeno,_)
- Dotazy: sestry(Ales,Maruska)

Operátory, definice.

Operátory – slouží pro zápis struktury, používají se hlavně v pravidlech

Struktura, funktor a arita, vysvětlete – příklad.

Struktura – funktor + arita (počet termů v závorce): osoba(jmeno,prijmeni,vek)

Použití seznamů, rekurzivní přístup při psaní v rámci logického programování.

//uziti seznamu v pravidle

slova(Sez) :- Sez = [jedna, dva, tri].

Popište postup prologu při vyhodnocování a plnění cílů.

Během procesu vyhodnocení cíle (dotazu) dochází v případě úspěchu k nastavení proměnných na takové hodnoty, které umožní jeho splnitelnost. Jeli více možností, jak dosáhnout splnění cíle, Prolog je schopen je všechny zjistit pomocí postupu nazývaného **zpětné prohledávání (backtracking)**.

Neúspěch při vyhodnocení cíle neznamena, že Prolog dokázal jeho neplatnost, ale pouze to, že nebyl schopen ze své databáze dokázat jeho platnost.

Popište operaci srovnání (unifikace) v Prologu.

Je to aktivní mechanismus, jehož výsledkem není jen odpověď ano/ne. Během procesu srovnání se totiž případně musí hledat vhodná substituce za proměnné, pomocí níž se podaří termy „zestejnit“. Dva termy se srovnají, když jsou identické, nebo když proměnné vyskytující se v těchto termech lze nastavit na takové hodnoty, že se získají identické termy.

Protože český termín srovnání má obecnější význam, používá se často místo něj termín unifikace. Výsledkem unifikace je vždy buď neúspěch, nebo substituce, s jejíž pomocí se termy unifikují, tj. stanou se identickými.

Popište blokový model vyhodnocování v Prologu.

MATLAB

Popis Matlabu a jeho využití.

MATLAB je matematický software určený pro vědecké a technické výpočty, modelování, zpracování signálů, analýzu a prezentaci dat.

MATrix LABoratory – hlavní je práce s poli (maticemi)

Definice skriptů za pomoci M souborů, tvorba podprogramů.

M-soubory jsou textové soubory určené pro ukládání posloupnosti příkazů (skripty), nebo pro ukládání uživatelských funkcí. Skripty jsou pomalejší než funkce (ty se předem zkompilují).

Skripty

Tvorba: File > New > Script nebo v libovolném txt editoru: nazev.m

Spuštění: v Command Window – nazev ; v Matlab editoru – Debug > Run (F5)

Zobrazení kódu v CW: příkazu type: >> type nazev

Funkce

Funkce jsou M-soubory začínající klíčovým slovem function. Mohou používat vstupní parametry a musí vracet nějakou hodnotu.

Soubor soucin.m:

```
function [s] = soucin(a,b)
% SOUCIN - soucin dvou cisel
s = a * b;
end
```

V konzoli zadám soucin(2,3) a vypíše se ans = 6

Sinus_graf.m:

```
function sinus_graf
% graf funkce sinus
t = 0:0.1:10 % vytvori vektor t o prvcich 0, 0.1 0.2 0.3 az 10
sinus = sin(t) % vektor sinus obsahuje hodnoty sinu pro t
plot(t,sinus) % vykresli graf
end
```

Tvorba proměnných, řádkových a sloupcových vektorů, matic.

Matlab navenek nerozlišuje různé typy proměnných – každá proměnná je matice.

Pro deklaraci proměnné se používá přiřazovací příkaz: nazev = hodnota; Řetězce: a = ,zdar'

Pole

Tvorba: a = [...] – sloupce se oddělují mezerou nebo čárkou, řádky středníkem; skalár: a = 7

Vektor – řádkový: a = [1 2 3]; sloupcový: a = [1;2;3], matice: a = [1 2 3; 4 5 6] ... dva řádky a tři sloupce

Přístup k prvku: a(m,n)

Operátory

Relační, logické – pro každou hodnotu v poli vrátí false nebo true

Aritmetické – unární, binární, dvojtečka – posloupnost ... i:k:j (výchozí k=1)

Vykreslování 2D/3D grafů a optimalizace zázornění.

Matlab obsahuje velmi dobrou podporu pro prezentaci dat. Ale jelikož pracuje s číslíkovou reprezentací funkcí, jsou data vždy diskrétní. Pokud je chceme zobrazit jako spojitě, použijeme dostatečně velkou hustotu nebo to aproximujeme (plot to dělá automaticky).

2D grafika

Vykreslovací funkce: plot (lineární stupnice), bar (sloupcový), stem (bodový)

Funkce pro popisky – přidá k vykreslenému grafu: title, xlabel, ylabel

```
% plot - jeden argument - jako y se doplní index z vektoru
t = 0:pi/100:2*pi;
plot(sin(t))
title('Sinus')
xlabel('t')
ylabel('y = sin(t)')
% plot - dva argumenty
x = [0,1,2,4,8,9,15];
y = [12,3,15,45,2,7,12];
plot(x,y)
title('Body 1')
xlabel('x')
ylabel('y')
```

Cykly, podmínky, chráněné bloky.

Podmínky

```
a = 3;
% IF
if a > 1
    disp('a je vetsi nez 1');
    % funkce dispoze vypise svůj argument na standardni vystup
    b = a + 1;
else
    disp('a je mensi nebo rovno 1');
end
% SWITCH - tohle mne teda nefunguje, ale nevim proc :)
msg = '';
switch a
    case 3 msg = 'je to spatny';
    case{4,5} msg = 'je to dobry';
    otherwise
        disp('jinak');
end
disp(msg);
```

Cykly

```
% FOR
for n = 1:10
    disp(n);
end
% WHILE
n = 2;
while n < 10
    disp(n);
    n = n + 1;
end
```

Otázkové okruhy bez zařazení

Programovací techniky pro umělou inteligenci: Sémantiky programovacích jazyků.

Logické programování a jazyk Prolog: řízení výpočtu, řez, seznamy, vestavěné predikáty, styl programování, optimalizační techniky.

Metody reprezentace znalostí a inference: reprezentace znalostí, pravidla, rámce, sémantické sítě.

Metody tvorby báze znalostí.

Prolog a reprezentace znalostí.

Hry a základní herní strategie.

Prohledávání a programování s omezujícími podmínkami: řešení problémů využitím vyhledávacích metod, prohledávání stavového prostoru.

Stromové prohledávání, lokální prohledávání, neúplné prohledávání.

Řešení optimalizačních problémů.

Dolování z dat a strojové učení: proces dobývání znalostí z dat a dolování z dat. Typické úlohy. Předzpracování dat. Strojové učení.

Neuronové sítě a genetické algoritmy.

Posilované učení (reinforcement learning).

Vizualizace dat.

Metody strojového učení pro zpracování textu.

Klasifikace a filtrace dokumentů. Extrakce informace z textu.