

# Téma 48 (dříve 47)

Martin Staviař, staviarm@centrum.cz

16. srpna 2006

Rozpoznávání a vnímání. Statistický (příznakový) a strukturní přístup. Klasifikátory a jejich učení. Cíle umělé inteligence. Reprezentace úloh, stavový prostor a jeho prohledávání. Logika a její využití při formalizaci řešení úloh umělé inteligence. Reprezentace znalostí. Heuristické znalosti. Expertní systémy a řízení. Znalostní inženýrství. Adaptivní a učící se algoritmy. Aplikace umělé inteligence.

## 1 Rozpoznávání a vnímání

Úloha rozpoznávání spočívá zařazování objektů, popř. jevů a situací reálného světa do tříd. Každý z objektů je při dostatečně podrobném popisu jedinečný, "třída" představuje zobecnění.

Je třeba nejprve stanovit hledisko, z něhož budou objekty posuzovány – určit veličiny, které jej charakterizují. Na objektu definujeme **systém** – volba dat a způsob, jak budou měřena (provádí řešitel úlohy na základě analýzy).

Získaná data jsou vstupními údaji pro rozpoznávání. Uspořádáme-li tato data do číselného vektoru, nazýváme tento vektor **obraz**. Svými hodnotami "zobrazuje" objekt z hlediska zvoleného systému. V dalším řešení s objektem nepracujeme, je již plně zastoupen obrazem.

Úloha rozpoznávání:

- zpracování dat naměřených na objektu (zpracování obrazu) tak, aby byla maximalizována diskriminační schopnost při minimalizaci dat,
- přiřazení indikátoru třídy jednotlivým popisům – vlastní klasifikace

## 2 Statistický (příznakový) a strukturní přístup

### 2.1 Příznakové metody rozpoznávání

Příznakové metody používají k popisu objektu hodnoty, které mají význam míry vlastnosti a nazývají se **příznaky**. Všechny příznaky, kterými popisujeme objekt, můžeme uspořádat do vektoru, který nazýváme **vektor příznaků**. Prostor všech těchto vektorů nazýváme příznakový prostor. Klasifikátor tedy zobrazuje příznakový prostor objektů na množinu indikátorů tříd. Následující části popisují příznakové metody rozpoznávání.

#### 2.1.1 Diskriminační funkce

Označme  $x_1, x_2, \dots, x_n$  příznaky,  $x^T = [x_1, \dots, x_n]$  je  $n$ -rozměrný vektor příznaků. Příznakový prostor je tedy  $n$ -rozměrný a označme jej  $\kappa$ . Předpokládejme klasifikaci do  $R$  tříd, indikátory tříd označíme  $w_1, w_2, \dots, w_R$ . Každému bodu  $x \in \kappa$  je přiřazen indikátor třídy  $w \in \{w_1, w_2, \dots, w_R\}$ . Funkce  $w = d(x)$  je rozhodovacím pravidlem, které popisuje toto přiřazení. Rozhodovací pravidlo v příznakovém prostoru  $\kappa$  vymezuje  $R$  vzájemně disjunktních podmnožin  $\kappa_1, \kappa_2, \dots, \kappa_r$ , definovaných vlastností

$\kappa_r \equiv \{x : d(x) = w\}$ . Nadplochy, které jsou společné dvěma množinám  $\kappa_i, \kappa_j$ , nazýváme rozdělující nadplochy. Jsou-li definovány rozdělující nadplochy, je definováno i rozhodovací pravidlo a úloha klasifikace je rozřešena.

Rozdělující nadplochy lze definovat pomocí  $R$  skalárních funkcí vektorového argumentu  $g_i(x), i = 1, \dots, R$ , které nazýváme **diskriminační funkce**. Každá z diskriminačních funkcí je přiřazena jedné z tříd. Diskriminační funkcí  $i$ -té třídy může být libovolná funkce  $g_i(x)$ , splňující nerovnost

$$g_i(x) > g_j(x)$$

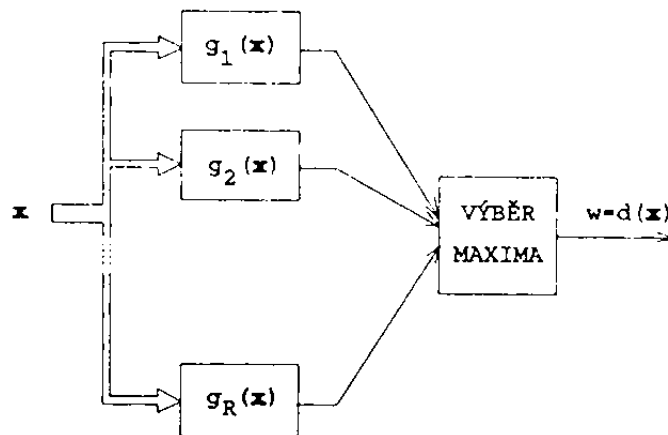
pro každý vektor  $x \in \kappa_i$  a pro  $j = 1, 2, \dots, R, j$  různé od  $i$ . Diskriminační funkce  $i$ -té třídy nabývá pro všechny vektory  $x \in \kappa_i$  větší hodnoty než diskriminační funkce ostatních tříd. Rovnice rozdělující nadplochy mezi sousedními množinami  $\kappa_i$  a  $\kappa_j$  mají tvar

$$g_i(x) = g_j(x).$$

Jestliže vektor leží na rozdělující ploše, nelze o jeho zařazení do třídy rozhodnout.

Klasifikátor vyhodnocuje pro zkoumaný vektor  $x$  hodnotu diskriminačních funkcí všech tříd a přiřazuje vektor  $x$  do té třídy, jejíž diskriminační funkce má největší hodnotu.

(Klasifikace do dvou tříd – *dichotomie* – stačí posuzovat znaménko rozdílu  $g(x) = g_1(x) - g_2(x)$ )



Obrázek 1: Klasifikátor popsaný diskriminačními funkcemi

### 2.1.2 Kritérium minimální vzdálenosti

Klasifikujeme do  $R$  tříd, každou charakterizuje vzorový vektor příznaků - **etalon** třídy. Klasifikovaný vektor pak řadíme do té třídy, jejíž etalon má od hledaného vektoru minimální vzdálenost.

### 2.1.3 Kritérium minimální chyby

Nastavujeme klasifikátor tak, aby ztráty způsobené chybným rozhodnutím byly minimální.

Pravděpodobnostně popsaná úloha, klasifikace do  $R$  tříd, indikátory  $w_1, w_2, \dots, w_R$ .  $w$  je indikátor třídy, do které patří vektor příznaků  $x$ . O hodnotě  $w$  neumíme jednoznačně rozhodnout, pokládáme ji za náhodnou proměnnou s možnými hodnotami  $w_1, w_2, \dots, w_R$  a s danými pravděpodobnostmi  $P(w_1), P(w_2), \dots, P(w_R)$ , pro které platí vztah

$$\sum_{i=1}^R P(w_i) = 1.$$

Hodnota  $P(w_i)$  (*apriorní pravděpodobnost*) může být například pravděpodobnost výskytu písmena v textu, nemoci v populaci.

Zařazujeme neznámý vektor příznaků  $x$  do třídy. Kromě apriorních pravděpodobností  $P(w_i)$  známe hodnotu vektoru  $x$  a všechny podmíněné hustoty pravděpodobností  $p(x/w_i)$ . Ty vyjadřují rozložení hodnot  $x$  v třídách. Pravděpodobnost, že vektor  $x$  patří do třídy s indikátorem  $w_j$ :

$$p(x) = \sum_{i=1}^R p(x/w_i)P(w_i)$$

Pravděpodobnost  $P(w_j/x)$  (*aposteriorní pravděpodobnost*). Pro její výpočet slouží *Bayesův vztah*

$$P(w_i/x) = \frac{p(x/w_i)P(w_i)}{p(x)};$$

Pomocí aposteriorních funkcí stanovíme rozhodovací pravidlo: Vektor  $x$  zařadíme do takové třídy  $j$ , pro kterou platí

$$P(w_j/x) = \max P(w_j/x).$$

Rozhodovací pravidlo lze zde vyjádřit také pomocí diskriminačních funkcí.

#### 2.1.4 Výběr a uspořádání znaků

Větší počet příznaků zvyšuje náklady na měření a nemusí vždy přinést užitek. Příznaky volí člověk na základě zkušeností, analýzy a často i intuice.

Při výběru příznaků musíme důsledně rozlišovat **extrakci** od **selekcce**. Při extrakci se příznaky vypočítávají z příznaků původních (založeno na Karhumenově-Loeově rozvoji). Extrahované příznaky postrádají fyzikální smysl. Nevýhodou je, že je nutné změřit všechny veličiny.

Při selekci se vybírá nejlepší možná podmnožina z původních proměnných bez transformace (založeno na míře diskriminativnosti). Zde si příznaky zachovávají význam. Výhodou je snížení počtu měřených veličin. Selekcí dosáhneme nižší diskriminační schopnosti než u extrakce.

## 2.2 Strukturální metody rozpoznávání

Metody strukturálního rozpoznávání používají s obrazy tvořenými souborem základních popisných elementů – **primitiv**, jejich vlastnostmi a relacemi mezi nimi. Primitiva představují minimální kvalitativní charakteristiky, které lze najít v obrazu. Relace mezi nimi jsou prostorové, funkční apod. Strukturální popis umožňuje řešit bohatší třídu úloh než příznakový popis. Umožňuje nahlédnout do struktury.

Na začátku úlohy se volí primitiva a relace (co nejmenší počet primitiv a relací, primitiva odpovídají přirozeným a výrazným strukt. elementům objektu, snaha o co nejjednodušší určení primitiv a relací).

Pro jednotlivé oblasti existují nejčastěji používaná primitiva a typické relace (dvojměrné obrazce – přímkové a křivkové úseky; trojměrné obrazce z dvojrozm. snímků – totožné typy projekcí vrcholů).

## 3 Klasifikátory a jejich učení

### Příklad:

Objekt – skupina lidí (každý je jedinečný – jméno, rodné číslo...).

Sledujeme je ze zdravotního hlediska – zda trpí či netrpí zvolenou srdeční chorobou. Tím definujeme dvě třídy: nemocní danou chorobou a ostatní.

Při posuzování využijeme záznam EKG (přesnost 5%, vzorkován s periodou 0,01s). Tím jsme definovali systém, ve kterém je každý pacient reprezentován  $n$ -ticí vzorků kardiogramu. Takto naměřená data ale

nejsou vhodná pro přímé použití klasifikační metody, proto volíme vhodnou metodu redukce objemu dat.

Klasifikace – pomocí **klasifikátoru** (algoritmus – zobrazuje množinu vektorů příznaků na množinu jmen (indikátorů) tříd – definuje **rozhodovací pravidlo**).

### 3.1 Učení s učitelem

Jedná se o sestavením rozhodovacího pravidla s použitím objektů, jejichž správná klasifikace je předem známa. Množina vektorů příznaků se známou klasifikací – **trénovací množina** – na ní se vytvoří klasifikátor. Potřeba nekonečně velké trénovací množiny je zastoupena pravděpodobnostním popisem vektorů příznaků a tříd.

### 3.2 Učení bez učitele

V případě, že máme k dispozici trénovací množinu obsahující pouze vektory příznaků bez toho, že by byly klasifikovány, jedná se o učení bez učitele. Jedním z prostředků učení bez učitele je *shluková analýza*, která umožňuje nastavení klasifikátoru bez údajů o správné klasifikaci, popřípadě i bez znalosti o počtu tříd.

Úkolem shlukové analýzy je, jak již vypovídá název, nalézt shluky neklasifikovaných vektorů množiny  $\tau$ . Shluky jsou definovány jako skupiny, jejichž prvky jsou si vzájemně blízké. Cílem je rozložit množinu  $\tau$  na co možná nejkompaktnější navzájem disjunkt ní podmnožiny.

## 4 Cíle umělé inteligence

Umělá inteligence je empirická věda, která se zabývá zkoumáním a chápáním inteligentních projevů. Nástrojem bádání je abstrakce a modelování inteligentních projevů mimo medium lidské mysli (zpravidla pomocí počítače). Cíle umělé inteligence nejlépe popisuje několik jejích definic:

- Minsky (1967): Umělá inteligence je věda o vytváření strojů nebo systémů, které budou při řešení určitého úkolu používat takového postupu, který – kdyby ho dělal člověk – bychom považovali za projev jeho inteligence.
- Richová, Knight (1991): Umělá inteligence se zabývá tím, jak počítačově řešit úlohy, které dnes zatím zvládají lidé lépe.
- Kotek (1983): Umělá inteligence je vlastnost člověkem uměle vytvořených systémů vyznačujících se schopností rozpoznávat předměty, jevy a situace, analyzovat vztahy mezi nimi a tak vytvářet vnitřní modely světa, ve kterých tyto systémy existují, a na tomto základě pak přijímat účelná rozhodnutí, za pomoci schopností předvídat důsledky těchto rozhodnutí a objevovat nové zákonitosti mezi modely nebo jejich skupinami.

## 5 Reprezentace úloh, stavový prostor a jeho prohledávání

Důležitým rysem inteligentních systémů je schopnost vytvářet si vnitřní strojový model prostředí (světa) a pracovat s ním. Systémy UI hledají vhodnou posloupnost akcí od počátečního modelu k cílovému. Každému modelu odpovídá jistý **stav** prostředí, množina všech stavů tvoří **stavový prostor**. Stavový prostor lze reprezentovat orientovaným grafem (uzel = stav, orientovaná hrana = přechod mezi stavy). Řešení úloh je hledáním přijatelné cesty v orientovaném grafu. Cílových stavů může být více a mohou být popsány pouze podmínkami, které mají splňovat.

## 5.1 Neinformované metody prohledávání

Slepé prohledávání do šířky a slepé prohledávání do hloubky. Při slepém prohledávání do šířky se nejprve expanduje uzel s minimální hloubkou. Postup je popsán následujícím algoritmem. (OPEN = seznam neexpandovaných stavů, CLOSED = seznam již expandovaných stavů).

Algoritmus prohledávání do šířky:

1. CLOSED = { }, OPEN = { $s_0$ }, kde  $s_0$  je počáteční stav. Je-li  $s_0$  současně i cílový stav, pak ukonči prohledávání.
2. Je-li OPEN prázdný, řešení neexistuje  $\rightarrow$  ukonči prohledávání!
3. Vyber a současně vymaž první stav ze seznamu OPEN, označ jej  $i$  a zapiš jej do seznamu CLOSED.
4. Expanduj stav  $i$ . Pokud stav  $i$  nemá následovníky nebo všichni následovníci byli již expandováni (t.j. jsou v seznamu CLOSED), jdi na krok 2.
5. Zapiš všechny následovníky stavu  $i$ , kteří nejsou v seznamu CLOSED na konec seznamu OPEN.
6. Pokud některý z následovníků stavu je cílovým stavem, tj. řešení bylo právě nalezeno, ukonči prohledávání, jinak pokračuj krokem č. 2.

Druhým základním algoritmem je slepé prohledávání do hloubky. Zde se nejprve expanduje uzel s největší hloubkou. Maximální hloubka může být omezena.

Algoritmus prohledávání do hloubky (s definovanou maximální hloubkou  $max$ ):

1. CLOSED = { }, OPEN = { $s_0$ }, kde  $s_0$  je počáteční stav. Je-li  $s_0$  současně i cílový stav, pak ukonči prohledávání.
2. Je-li OPEN prázdný, řešení neexistuje  $\rightarrow$  ukonči prohledávání!
3. Vyber a současně vymaž první stav ze seznamu OPEN, označ jej  $i$  a zapiš jej do seznamu CLOSED.
4. Pokud se hloubka uzlu  $i$  rovná maximální přípustné hloubce  $max$ , pokračuj krokem 2.
5. Expanduj stav  $i$ . Pokud stav  $i$  nemá následovníky nebo všichni následovníci byli již expandováni (t.j. jsou v seznamu CLOSED), jdi na krok 2.
6. Zapiš všechny následovníky stavu  $i$ , kteří nejsou v seznamu CLOSED na začátek seznamu OPEN.
7. Pokud některý z následovníků stavu je cílovým stavem, tj. řešení bylo právě nalezeno, ukonči prohledávání, jinak pokračuj krokem č. 2.

Výhodou prohledávání do hloubky jsou nižší nároky na paměť, neboť se v ní uchovávají pouze uzly na cestě od počátečního stavu ke stavu právě expandovanému. Oproti tomu výhodou prohledávání do šířky je fakt, že vždy vede k nalezení nejkratší cesty.

## 5.2 Informované metody prohledávání

Je možné definovat **hodnotící funkci**  $f$  (poskytuje odhad kvality stavu vzhledem k řešené úloze – čím nižší hodnota, tím blíže řešení), která určuje ohodnocení uzlů. Hodnoty hodnotící funkce se používají při výběru uzlu pro expanzi. Zabrání prohledávání cest, které nevedou k cíli.

**Gradientní algoritmus** vybírá k expanzi vždy ten uzel  $u$ , který  $f$  hodnotí jako nejslibnější, jeho rodiče i sourozence zapomíná. Končí, když nalezeno řešení nebo když všichni následníci uvaž. uzlu  $u$  mají horší ohodnocení  $f$  než  $u$ .

Účinnější je **algoritmus uspořádaného prohledávání** - vznikl rozšířením gradientního algoritmu o paměť.

## 6 Logika a její využití při formalizaci řešení úloh umělé inteligence

Zabývá se metodami přesvědčivé argumentace, studiem a formalizací této části zpracování informací. Metody logiky jsou nezbytné především při vysvětlování a odvozování nových tvrzení.

**Výroková logika** - formální odvozovací systém, ve kterém atomické formule tvoří výrokové proměnné (narozdíl od predikátové logiky).

**Predikátová logika** - formální odvozovací systém používaný k popisu matematických teorií a vět. Predikátová logika je rozšířením výrokové logiky (ta nedokáže vyjádřit některá složitější tvrzení o matematických strukturách). Do této logiky přidává kvantifikátory a vztah predikát – individuum. Individuum je prvek z nějaké množiny a predikát je relace na této množině.

Logické spojky:

není pravda, že;  $\neg$  *negace*

a;  $\wedge$  *konjunkce*

nebo;  $\vee$  *disjunkce*

jestliže ... , pak;  $\Rightarrow$  *implikace*

právě tehdy, když;  $\Leftrightarrow$  *ekvivalence*

Formální systém:

**Axiom** = základní tvrzení, které se považuje za pravdivé, aniž by k němu byl požadován důkaz.

1.  $\alpha \rightarrow (\beta \rightarrow \alpha)$
2.  $(\alpha \rightarrow (\beta \rightarrow \gamma)) \rightarrow ((\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \gamma))$
3.  $(\neg\beta \rightarrow \neg\alpha) \rightarrow (\alpha \rightarrow \beta)$

**Odvozovací pravidla:**

*Modus ponens* – dovoluje ze dvou formulí tvaru  $\alpha$  a  $\alpha \rightarrow \beta$  odvodit formuli  $\beta$ .

Pravidlo generalizace – „Pro libovolnou proměnnou  $X$  odvoď z formule  $\alpha$  formuli  $\forall X\alpha$ ”

## 7 Reprezentace znalostí

Inteligentní systém musí umět předvídat důsledky svých akcí - potřebuje *model svého prostředí*. K jeho konstrukci potřebuje **znalosti**:

- Deklarativní – např. logika 1. řádu, explicitně uvedené informace lze lehce upravovat, podporuje doplňování důsledků
- Procedurální – např. jízda na kole – informace jsou implicitně použité v procedurách realizujících nějakou úlohu

Požadavky na reprezentaci znalostí pro systémy UI: Modifikovatelnost, Modularita (funkčně souvislé části tvoří samostatné části: např. produkční systémy), Sémantické sdružování znalostí (vhodné řazení do tříd a jejich hierarchií).

Znalosti jsou získávány přímo pozorováním reálných předmětů a jevů a nepřímo odvozováním. Každé poznávání je vázáno na jazyk, jímž vypovídáme o té části světa, kterou zkoumáme. Pro popis

znalostí daného oboru používáme umělé jazyky (matematické formule, chemické značky, programovací jazyky, schemata elektrických obvodů).

Produkční systém = soubor produkčních pravidel (Situace  $\rightarrow$  Akce), počáteční i odvozená data jsou uložena v pracovní paměti (báze dat) – popisuje okamžitý stav řešené úlohy. Inferenční stroj porovnává data v pracovní paměti s produkčními pravidly, vybírá vhodná pravidla a provádí akci. **Přímé řetězení** – odvozované řízené daty, začíná ve výchozím stavu; **Zpětné řetězení** – řízené cílem (PROLOG).

Sémantické sítě = Znalosti jsou reprezentovány pomocí **objektů** a **relací** mezi nimi, pomocí grafu (uzel = objekt, hrana = binární relace). Rozlišují se typy a instance typu. Je kladen důraz na dědění.

Rámce = Další schema reprezentace znalostí, oproti sémantickým sítím používají tabulky. Obsahují **položky** (slouží k popisu jednotlivých vlastností objektu). V průběhu užívání rámce nabývají položky konkrétních hodnot. Popis položky se skládá ze **jména** a **hodnoty**. Položky se dále dělí na **fasety**.

## 8 Heuristické znalosti

Heuristické znalosti (heuristiky) mají empirický charakter, mohou to být neexaktní poznatky, o nichž víme, že jsou často užitečné při řešení. Heuristiky se používají tam, kde není k dispozici exaktní algoritmus. Mohou mít různou podobu, lepší soubor heuristik zajišťuje prohledávání menší části stavového prostoru, přímočařejší postup a způsob řešení se jeví jako inteligentnější.

## 9 Expertní systémy a řízení

Expertní systémy jsou počítačové programy simulující rozhodovací činnost specialistů (expertů) při řešení složitých úloh rozhodování. Využívají vhodně zakódované speciální znalosti převzaté od expertů s cílem dosahovat ve zvolené problémové oblasti kvality rozhodování na úrovni experta. Znalosti převzaté od experta tvoří **bázi znalostí**, která je obvykle implementována, spravována a udržována jako samostatný soubor. Báze znalostí značně ovlivňuje efektivitu ES. *Teorie zpracování neurčitosti*, důležitá oblast teorie UI, která se vyrovnává s nepřesnými znalostmi.

První prokazatelně úspěšné diagnostické expertní systémy – 70. léta, MYCIN (infekční onemocnění krve), PROSPECTOR (geologie), pracují s neurčitostí. Komerční nasazení expertních systémů přineslo značné finanční úspory

Typy ES:

- Diagnostické – Mají za úkol porovnávat a vyhodnocovat předem stanovené hypotézy. Cílem je určit, která z těchto hypotéz nejlépe odpovídá reálným datům.
- Plánovací – Při řešení úloh v plánovacím expertním systému je znám cíl řešení a počáteční stav. Úkolem systému je s využitím dat o daném případě nalézt optimální posloupnost kroků (operátorů), kterými lze dosáhnout stanoveného cíle.
- Hybridní – Kombinací architektury diagnostického a plánovacího systému vzniká hybridní expertní systém. Tímto typem jsou například inteligentní výukové systémy či monitorovací systémy.

## 10 Znalostní inženýrství

V souvislosti s vývojem expertních systémů vznikl v rámci umělé inteligence samostatný obor, který dostal název znalostní inženýrství (knowledge engineering). Znalostní inženýrství se obecně zabývá tvorbou expertních systémů, jejich aplikací, údržbou a integrací s jinými softwarovými produkty. Nejvýznamnější činnosti souvisejí s naplňováním expertních systémů znalostmi (metody a techniky získávání znalostí, jejich formalizace, kódování, uchovávání, testování a udržování). Kromě vlastní

práce se znalostmi se znalostní inženýrství zabývá tříděním a katalogizací dostupných metod a technik reprezentace znalostí, inferenčních strojů, vysvětlovacích mechanismů, prostředků počítačové podpory návrhu expertních systémů a tvorbou relevantních metodologií.

Spolu s novým oborem vznikla také nová profese - **znalostní inženýr**. Znalostní inženýr musí být seznámen s problematikou umělé inteligence a expertních systémů, s technickými možnostmi reprezentace znalostí a s dostupnými inferenčními stroji. Kromě toho se musí před tvorbou expertního systému (resp. báze znalostí) podrobně seznámit s terminologií a základy problémové oblasti, získávat od experta znalosti v průběhu celého procesu tvorby báze znalostí, formulovat tyto znalosti způsobem vhodným pro počítačovou reprezentaci a kódovat je do tvaru vhodného pro daný expertní systém.

## 11 Adaptivní a učící se algoritmy

Jedná se o takové algoritmy, které jsou schopny se přizpůsobit změně. Korigují vyhodnocování dle aktuálních podmínek (spam filtry, šachy - algoritmus učící se od hráče).

## 12 Aplikace umělé inteligence

Počítačové vidění, Expertní systémy, Strojové učení, Zpracování přirozeného jazyka, Strojový překlad, Robotika, Diagnostika...

## Reference

- [1] Mařík V., Štěpánková O., Lažanský J. *Kybernetika a umělá inteligence 1*. Academia (1993)
- [2] Demlová M., Pondělíček B. *Matematická logika*. ČVUT (1999)