

Adaptivní inteligentní systémy

Petr Berka
Laboratoř inteligentních systémů
VŠE, Praha
berka@vse.cz

Osnova

- historické ohlédnutí
- strojové učení, soft computing a adaptivita
- adaptivita v metodách strojového učení
 - inkrementální učení
 - integrace znalostí
 - meta učení
 - revize znalostí
 - učení a zapominání
 - využití analogií

Znalosti 2003, Petr Berka

2

Adaptivní a učící se systémy v teorii řízení

- vstupy
 - projevy prostředí $\mathbf{x} \in X$
 - informace o požadovaném chování systému $\Omega \in O_1$
- výstup
 - reakce systému $\omega \in O_2$ na dvojici $[\mathbf{x}, \Omega]$
reakce je dána funkcí f takovou, že $\omega = f(\mathbf{x}, q)$, kde q je parametr systému

každá reakce systému může způsobit ztrátu $Q(\mathbf{x}, \Omega, q)$

Znalosti 2003, Petr Berka

3

Adaptivní systém

v průběhu **adaptrace** systém pro každou dvojici $[\mathbf{x}, \Omega]$

hledá takový parametr q^* , pro který

$$Q(\mathbf{x}, \Omega, q^*) = \min_q Q(\mathbf{x}, \Omega, q)$$

Znalosti 2003, Petr Berka

4

Učící se systém

v průběhu **učení** systém na základě trénovací množiny

$$[\mathbf{x}_k, \Omega_k]$$

hledá takový parametr q^* , pro který

$$J(q^*) = \min_q J(q)$$

kde $J(q)$ je střední ztráta

$$J(q) = \int Q(\mathbf{x}, \Omega, q) dF(\mathbf{x}, \Omega)$$

(Kotek a kol., 1980)

Znalosti 2003, Petr Berka

5

Adaptivní inteligentní systémy

- schopnost přizpůsobit se změnám prostředí,
- schopnost přizpůsobit se novým podmínkám využívání,
- schopnost přizpůsobit se nové aplikaci.

[IST-2000-29270 Projekt EUNITE]

Znalosti 2003, Petr Berka

6

Soft computing

- „Methods that exploit the tolerance for imprecision, uncertainty and partial truth to achieve tractability, robustness and low solution cost.“

(Zadeh, 1994)

- fuzzy logika
- neuronové sítě
- genetické algoritmy
- pravděpodobnostní usuzování
- teorie chaosu

Znalosti 2003, Petr Berka

7

Strojové učení (1/2)

- „The field of machine learning is concerned with the question of how to construct computer programs that automatically improve with experience.“

(Mitchell, 1997)

- „Things learn when they change their behavior in a way that makes them perform better in a future.“

(Witten, Frank, 1999)

Znalosti 2003, Petr Berka

8

Strojové učení (2/2)

- učení se dovednostem
 - zdokonalování schopností na základě procvičování (např. hledání cesty v bludišti)
- učení se konceptům
 - hledání popisů tříd objektů, obecných zákonitostí (např. rozpoznání defraudanta)
 - s učitelem
 - bez učitele

Znalosti 2003, Petr Berka

9

Metody strojového učení

- tvorba rozhodovacích stromů
- tvorba rozhodovacích pravidel
- tvorba asociačních pravidel
- neuronové sítě
- genetické algoritmy
- bayesovské sítě
- učení založené na analogii
- induktivní logické programování

Znalosti 2003, Petr Berka

10

Adaptivita ve strojovém učení

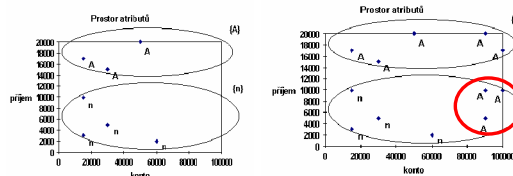
- inkrementální učení
 - integrace znalostí
 - meta učení
 - revize znalostí
 - učení a zapomínání
 - využití analogií
- Adaptivita 1
 - Adaptivita 1 a 2
 - Adaptivita 3

Znalosti 2003, Petr Berka

11

Učení se konceptům

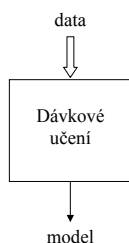
- objekty, patřící do téže třídy mají podobné charakteristiky (**učení na základě podobnosti**)
- z konečného počtu příkladů odvozujeme obecné znalosti (**induktivnost**)



Znalosti 2003, Petr Berka

12

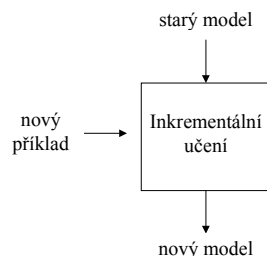
Dávkové učení



Znalosti 2003, Petr Berka

13

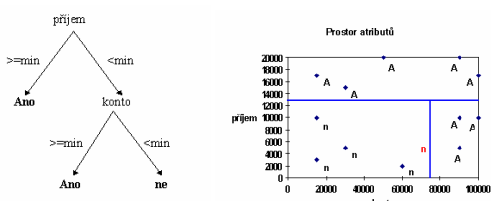
Inkrementální učení



Znalosti 2003, Petr Berka

14

Rozhodovací stromy



Znalosti 2003, Petr Berka

15

Rozhodovací stromy – základní dávkový algoritmus

Algoritmus TDIDT (ID3)

1. zvol jeden atribut jako kořen dílčího stromu,
2. rozděl data v tomto uzlu na podmnožiny podle hodnot zvoleného atributu a přidej uzel pro každou podmnožinu,
3. existuje-li uzel, pro který nepatří všechna data do téže třídy, pro tento uzel opakuj postup od bodu 1, jinak skonči.

(např. Quinlan, 1986)

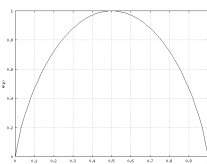
Znalosti 2003, Petr Berka

16

Rozhodovací stromy – volba atributu

entropie

$$H = - \sum_{i=1}^T (p_i \times \log_2 p_i)$$



Informační zisk, Gini index, χ^2 , ...

Znalosti 2003, Petr Berka

17

Rozhodovací stromy – inkrementální varianta (1/1)

ID4 (Schlimmer, Fisher, 1986)

1. na základě nového příkladu nalezni nejvhodnější atribut pro větvení
2. není-li nalezený atribut použit pro větvení
 - 2.1 odstraň celý podstrom
 - 2.2 použij vybraný atribut pro větvení (rozděl data na podmnožiny a přidej uzly do stromu)
3. existuje-li uzel, pro který nepatří všechna data do téže třídy, pro tento uzel opakuj postup od bodu 1, jinak skonči.

Znalosti 2003, Petr Berka

18

Rozhodovací stromy – inkrementální varianta (1/2)

ID5 (Utgoff a kol., 1997)

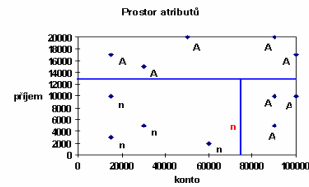
1. aplikuj starý strom na nový příklad
2. pokud se třída v listovém uzlu liší od třídy příkladu potom vytvoř z listu nový podstrom
3. pro všechny nelistové uzly v cestě použité pro nový příklad
 - 3.1 pokud atribut použitý pro větvení není po zahrnutí nového příkladu nejlepší
 - 3.1.1 použij nejlepší atribut
 - 3.1.2 rekurzivně transformuj celý podstrom

Znalosti 2003, Petr Berka

19

Rozhodovací pravidla

```
IF příjem >= min_příjem THEN Ano
IF příjem < min_příjem & konto < min_konto THEN ne
IF příjem < min_příjem & konto >= min_konto THEN Ano
```



Znalosti 2003, Petr Berka

20

Rozhodovací pravidla – základní charakteristiky

- Syntaxe
IF Ant THEN Class, Ant je konjunkce kategorií

- Charakteristiky

	Class	¬Class
Ant	a	b
¬Ant	c	d

- spolehlivost
 $a/(a+b)$

- pokrytí
 $a/(a+c)$

Znalosti 2003, Petr Berka

21

Rozhodovací pravidla – základní dávkový algoritmus

Algoritmus pokrývání množin

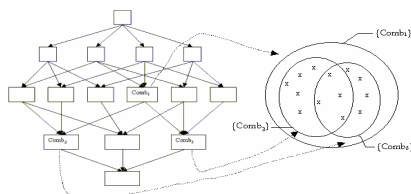
1. najdi pravidlo, které pokrývá nějaké pozitivní příklady a žádný negativní,
2. odstraň pokryté příklady z trénovací množiny,
3. pokud v trénovací množině zůstávají nějaké nepokryté pozitivní příklady, vrať se k bodu 1, jinak skonči.

Pokrývání „zdola nahoru“ (např. AQ – Michalski, 1969) nebo „shora dolů“ (např. CN2 – Clark, Niblett, 1989)

Znalosti 2003, Petr Berka

22

Učení jako prohledávání



- od obecnějšího modelu ke speciálnějšímu (*specializace*)
- od speciálnějšího modelu k obecnějšímu (*generalizace*)

Znalosti 2003, Petr Berka

23

Rozhodovací pravidla – inkrementální varianta (1/2)

YAILS (Torgo, 1992)

1. Pro každé existující pravidlo
 - 1.1 pokud nový příklad vyhovuje předpokladu i závěru pak nedělej nic
 - 1.2 pokud nový příklad vyhovuje pouze předpokladu pak hledej specializaci pravidla
 - 1.3 pokud nový příklad vyhovuje pouze závěru pak hledej generalizaci pravidla
2. Pokud nový příklad není pokryt žádným pravidlem, najdi nové pravidlo

Znalosti 2003, Petr Berka

24

Rozhodovací pravidla – inkrementální varianta (2/2)

PRIMEROSE-INC (Tsumoto, Tanaka, 1997)

- Pro každou kategorii A(v) pokud pokud klesne její pokrytí po vstupu nového příkladu, zařaď A(v) do seznamu K jinak zařaď A(v) do seznamu R
- Pro každou kategorii A(v) ze seznamu K najdi v seznamu „Vyhov“ pravidlo, které obsahuje A(v) a které nyní nevyhovuje požadavkům na spolehlivost a pokrytí a přesuň je do seznamu „Nevyhov“
- Pro každou kategorii A(v) ze seznamu R najdi v seznamu „Nevyhov“ pravidlo, které obsahuje A(v)
 - 3.1 pokud toto pravidlo nyní vyhovuje požadavkům na spolehlivost a pokrytí a přesuň je do seznamu „Vyhov“ jinak hledej jeho vyhovující specializaci
- Pokud ostatní pravidla z „Nevyhov“ nyní vyhovují, přesuň je do „Vyhov“

Znalosti 2003, Petr Berka

25

Učení založené na instancích – IBL (1/2)

- klasifikování nových příkladů na základě minima vzdálenosti od instancí v databázi (k-NN algoritmus)

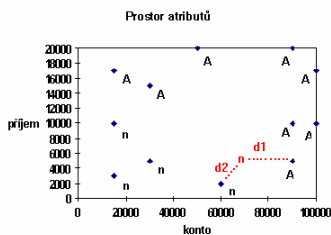
$$d(x_1, x_2) = \sqrt{\sum_{j=1}^m (x_{1j} - x_{2j})^2} \quad d(x_1, x_2) = \sum_{j=1}^m d(x_{1j}, x_{2j}), \text{ kde } d(x_{1j}, x_{2j}) = \begin{cases} 0 & \text{prox}_{x_{1j}} = x_{2j} \\ 1 & \text{prox}_{x_{1j}} \neq x_{2j} \end{cases}$$

- volba instancí pro uložení do databáze (Aha, 1991)
 - IB1 - systém ukládá každý příklad z trénovacích dat
 - IB2 - systém ukládá příklady chybně zařazené stávajícími instancemi
 - IB3 - kritérium pro volbu příkladů založené na souhrnné správnosti klasifikace

Znalosti 2003, Petr Berka

26

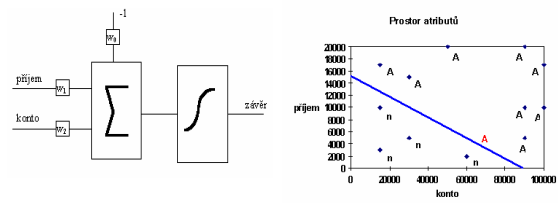
Učení založené na instancích – IBL (2/2)



Znalosti 2003, Petr Berka

27

Neuronové sítě - jeden neuron (1/2)



Znalosti 2003, Petr Berka

28

Neuronové sítě – jeden neuron (2/2)

- Dávková verze
- Inkrementální verze

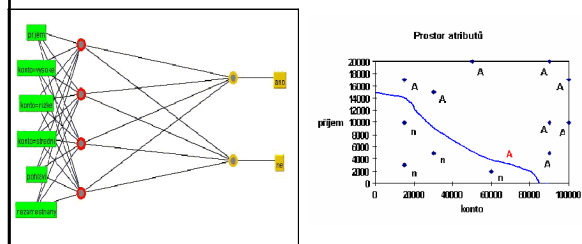
- náhodně inicializuj váhy
- dokud není splněno kritérium pro zastavení
 - 2.1 zpracuj všechny trénovací příklady $[x_i, y_i]$
 - 2.2 aktualizuj váhy

$$\Delta w = \eta \sum_i (y_i - \hat{y}_i) x_i$$

Znalosti 2003, Petr Berka

29

Neuronové sítě - vícevrstvá síť



Znalosti 2003, Petr Berka

30

Backpropagation

1. inicializuj váhy sítě malými náhodnými čísly
2. dokud není splněno kritérium pro zastavení
 - 2.1. pro každý příklad $[x, y]$ z trénovacích dat
 - 2.1.1. spočítej výstup out_u pro každý neuron u v síti
 - 2.1.2. pro každý neuron v ve výstupní vrstvě spočítej chybu $error_v = out_v (1 - out_v) (y_v - out_v)$
 - 2.1.3. pro každý neuron s ve skryté vrstvě spočítej chybu $error_s = out_s (1 - out_s) \sum_{v \in \text{výstup}} (w_{s,v} error_v)$
 - 2.1.4. pro každou vazbu vedoucí od neuronu j do neuronu k modifikuj váhu vazby

$$w_{j,k} = w_{j,k} + \Delta w_{j,k} \quad \text{kde } \Delta w_{j,k} = \eta error_k x_{j,k}$$

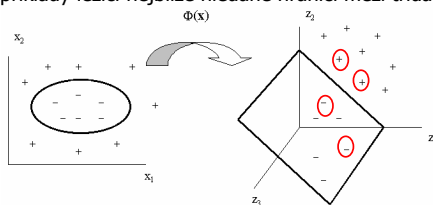
Sítě s kaskádovou architekturou

- architektura
 - 3 vrstvy: výstupní neurony spojeny se všemi neurony ze skryté i vstupní vrstvy, neurony ze skryté vrstvy spojeny se všemi neurony ze vstupní vrstvy i s „předcházejícími“ neurony ve skryté vrstvě (bráno např. zleva doprava)
- učení
 - konstruktivní inkrementální algoritmus: v průběhu učení se postupně přidávají neurony do skryté vrstvy, síť lze doučovat

(Fahlman, Lebiere, 1989)

Metoda SVM (1/2)

- Převod úlohy klasifikace tříd, které nejsou lineárně separabilní v původním prostoru atributů na úlohu lineárně separabilní ve vícozměrném prostoru
- Rozhodující pro nalezení rozdělovací nadroviny jsou příklady ležící nejbliže hledané hranici mezi třídami



Metoda SVM (2/2)

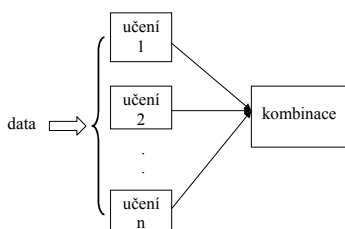
- Dávková verze
- Inkrementální verze

Pro učení použít všechna data (např. Cortes, Vapnik, 1995)

Pro učení použít staré podpůrné vektory (vážené počtem starých příkladů na jeden podpůrný vektor) a nová data

(Ruping, 2001)

Integrovaní znalostí na úrovni usuzování



- Bagging
- Boosting
- Stacking

(Bauer, Kohavi, 1999),
(Diettrich, 2000)

Bagging

- učení - vytvoření „paralelních“ modelů
 - vytvoř několik trénovacích množin (náhodný výběr s opakováním)
 - použij (týž) algoritmus pro učení
- klasifikace
 - rovné hlasování modelů

Boosting

- učení - vytvoření sekvence modelů
 - každý model se zaměřuje na ty příklady, které se předcházejícími modely nepodařilo klasifikovat (vážení příkladů)
 - použít (týž) algoritmus pro učení
- klasifikace
 - vážené hlasování modelů

AdaBoost

učení

1. přiřad' stejnou váhu všem příkladům,
2. pro každou iteraci
 - 2.1. vytvoř model
 - 2.2. spočítej chybu err
 - 2.3. Je-li $err=0$ nebo $err \geq 0.5$ konec
 - 2.4. pro každý příklad je-li klasifikace správně pak váha $w=w*err/(1-err)$
 - 2.5. normalizuj váhy příkladů

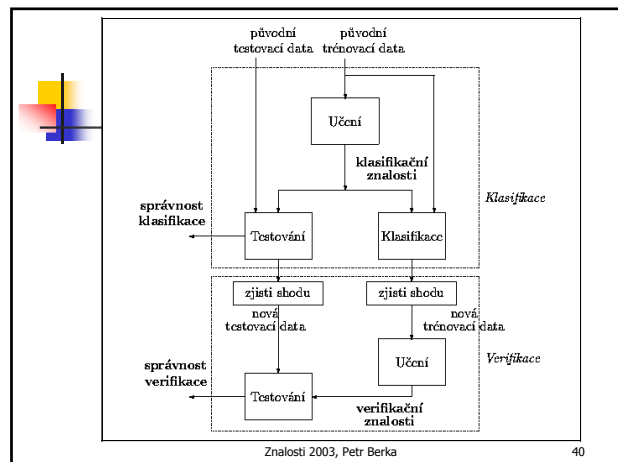
klasifikace

1. přiřad' váhy 0 všem třídám
2. pro každý model přiřad' třídě určené modelem váhu $w=w \cdot \log(err/(1-err))$
3. vydej třídu s nejvyšší váhou

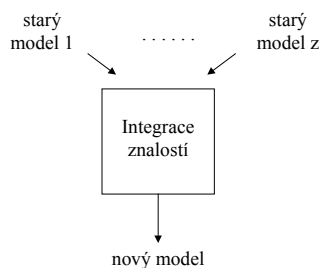
(Shapire, 1999)

Stacking

- učení - vytvoření „paralelních“ modelů
 - použít různé algoritmy na stejná data
- meta-učení (učení z výsledků jednotlivých klasifikací)
 - kombinování modelů
 - selekce modelů (Bensusan a kol., 2000)
- určení „oblasti expertízy“ modelu (Berka, 1997b)



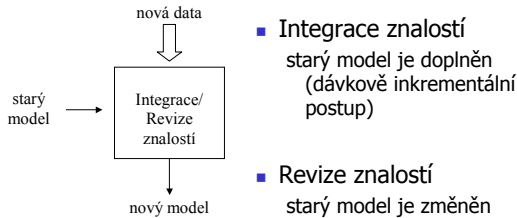
Integrovaní znalostí na úrovni reprezentace (1/2)



Integrovaní znalostí na úrovni reprezentace (2/2)

- výlučná věc znalostního inženýrství
- kombinace znalostního inženýrství a strojového učení
 - strojové učení jako doplňkový nástroj (Ganascia a kol., 1993 nebo Graner, Sleeman, 1993)
 - tvorba diagnostických pravidel indukci z příkladů, dedukci z teorie a abdukci z kausálních modelů (Saitta a kol., 1993)

Integrace/Revize znalostí



Induktivní logické programování (ILP)

- Úloha ILP (Lavrač, Džeroski, 1994 nebo Muggleton 1992):
 - Je dána množina Hornových klauzulí B (teorie), množina pozitivních příkladů EX^+ a množina negativních příkladů EX^- . Najdi hypotézu H takovou, že:
 1. $\forall ex \in EX^+ \quad H \wedge B \vdash ex$
 2. $\forall ex \in EX^- \quad H \wedge B \not\vdash ex$
 3. $H \wedge B \not\vdash \square$
- **výhody ILP**
 - možnost řešit úlohy v oblastech vyžadujících reprezentaci více relací (více propojených tabulek, strukturální data)
 - možnost využít doménových znalostí

Tvorba rozhodovacích pravidel algoritmem KEX

- **báze znalostí**
 pravidla $Ant \Rightarrow Class(w)$
- **inferenční mechanismus**
 Příspěvky všech aplikovatelných pravidel se složí jako

$$w_1 \oplus w_2 = \frac{w_1 \times w_2}{w_1 \times w_2 + (1 - w_1) \times (1 - w_2)}$$
- **způsob tvorby pravidel**
 Do báze pravidel se zařazují jen ty implikace, které nejsou odvoditelné z již získaných kratších pravidel, tedy pro které $P(Class|Ant)$ se liší od $cw(Ant)$

KEX dávková verze

- necht' CAT je seznam kategorií $A(v)$
 necht' $IMPL$ je seznam implikací $A(v) \Rightarrow Class$
 (oba seznamy jsou uspořádány sestupně dle četnosti $n(A(v))$)
 necht' báze pravidel KB obsahuje pouze prázdný vztah $\emptyset \Rightarrow Class(w)$
- dokud $IMPL$ není prázdný seznam
1. vezmi první implikaci ze seznamu $IMPL$ (označ ji $Ant \Rightarrow Class$)
 2. spočítej váhu $cw(Ant)$
 3. pokud se $P(Class|Ant)$ na základě χ^2 testu liší od $cw(Ant)$ potom
 - 3.1 přidej do KB pravidlo $Ant \Rightarrow Class(w)$, kde $w \oplus cw(Ant) = P(Class|Ant)$
 - 3.2 prodluž Ant o všechny vhodné kategorie $A(v)$ z CAT a zařaď $Ant \wedge A(v) \Rightarrow Class$ podle četnosti do seznamu $IMPL$
 4. odstraň $Ant \Rightarrow Class$ ze seznamu $IMPL$

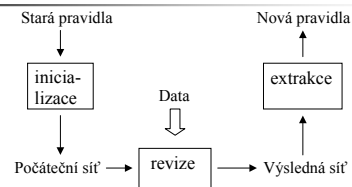
(Berka, Ivánek, 1994)

KEX verze pro revizi znalostí

- necht' KB je již existující báze pravidel
 necht' $IMPL$ je seznam implikací $Ant \Rightarrow Class$ vytvořených z nových dat uspořádaný sestupně dle četnosti $n(Ant)$
- dokud $IMPL$ není prázdný seznam
1. vezmi první implikaci ze seznamu $IMPL$ (označ ji $Ant \Rightarrow Class$)
 2. spočítej váhu $cw(Ant)$
 3. pokud se $P(Class|Ant)$ na základě χ^2 testu liší od $cw(Ant)$ potom
 - 3.1 přidej/přešpiš do KB pravidlo $Ant \Rightarrow Class(w)$
 - 3.2 pro každé pravidlo $Comb_r \Rightarrow Class(w_r)$ z KB takové, že Ant je podkombinací kombinace $Comb_r$
 - 3.2.1 spočítej váhu $cw(Comb_r)$
 - 3.2.2 pokud se $P(Class|Comb_r)$ signifikantně liší od $cw(Comb_r)$ potom modifikuj váhu w_r , jinak odstraň pravidlo $Comb_r \Rightarrow Class(w_r)$ z KB
 4. odstraň $Ant \Rightarrow Class$ ze seznamu $IMPL$

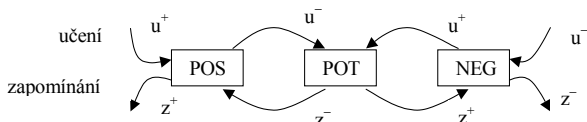
(Berka, 1997)

Revize pravidel pomocí neuronových sítí



- **bez změny struktury**
 - KBANN (Towell, Shavlik, 1993)
 - (Berka, Sláma, 1998)
- **se změnou struktury**
 - RAPTURE (Mahoney, Mooney, 1994)

Učení a zapominání



- s učitelem
 - FLORA (Widmer, Kubát, 1996)
 - STAGGER (Schlimmer, Granger, 1986)
- bez učitele
 - COBWEB (Fisher, 1987)

Znalosti 2003, Petr Berka

49

Učení...

Pro každý příklad σ_i z trénovací množiny

1. Je-li σ_i pozitivní příklad potom
 - 1.1 Pro každou $Comb$ z POS pokud $Comb$ pokrývá σ_i přiřaď $n_+(Comb) := n_+(Comb) + 1$
 - 1.2 Pro každou $Comb$ z POT pokud $Comb$ pokrývá σ_i přiřaď $n_+(Comb) := n_+(Comb) + 1$
 - 1.3 Pro každou $Comb$ z NEG pokud $Comb$ pokrývá σ_i přiřaď $n_+(Comb) := 1$ a přesuň $Comb$ do POT
 - 1.4 Pokud v POS není žádná $Comb$ která pokrývá σ_i , přidej do POS novou kombinaci $CombN$, která pokryje σ_i a nebude (kvůli sporu) v souladu s hypotézami v POT a NEG , a přiřaď $n_+(CombN) := 1$
2. Je-li σ_i negativní příklad potom postupuj analogicky....

(Widmer, Kubát, 1996)

Znalosti 2003, Petr Berka

50

... a zapominání

Pro každý příklad σ_i z trénovací množiny

1. je-li σ_i pozitivní příklad, potom
 - 1.1 pro každou $Comb$ z POS
 - 1.1.1 pokud $Comb$ pokrývá σ_i , přiřaď $n_+(Comb) := n_+(Comb) - 1$
 - 1.1.2 je-li $n_+(Comb) = 0$, odstraň $Comb$ z POS
 - 1.2 pro každou $Comb$ z POT
 - 1.2.1 pokud $Comb$ pokrývá σ_i , přiřaď $n_+(Comb) := n_+(Comb) - 1$
 - 1.2.2 je-li $n_+(Comb) = 0$, přesuň $Comb$ do NEG
2. Je-li σ_i negativní příklad potom postupuj analogicky....

(Widmer, Kubát, 1996)

Znalosti 2003, Petr Berka

51

Koncepty závislé na kontextu

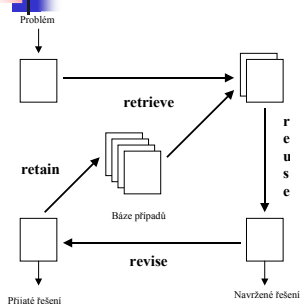
- Kontext
 - situace, ve které získáváme data
 - relevantní atributy, které nejsou v současnosti dostupné
 - atributy, které samy o sobě nepřispívají ke klasifikaci ale které zlepšují výsledky klasifikace v kombinaci s jinými atributy

(Matwin, Kubát, 1996)

Znalosti 2003, Petr Berka

52

Analogie a adaptace: Case-Based Reasoning



- retrieve (najdi nejpodobnější případy)
- reuse (použij navržené řešení)
- revise (případně reviduj navržené řešení)
- retain (uchovej nové řešení v bázi případů = učení)

(Watson, Marir, 1994)

Znalosti 2003, Petr Berka

53

Závěr

Pojem adaptivní inteligentní systémy bývá v současnosti úzce spojován s pojmem soft computing. V této oblasti reprezentují neuronové sítě a genetické algoritmy strojové učení v širším slova smyslu. Cílem tutorialu bylo ukázat, že i další (symbolické) metody učení se konceptům mohou vyhovovat různým aspektům definice adaptivity a vykazovat tedy rysy adaptivního chování.

Znalosti 2003, Petr Berka

54

Literatura (1/6)

- Aamodt, A. - Plaza, E. (1994): Case-based reasoning: Foundational issues, methodological variations, and system approaches. *Artificial Intelligence Communications*, 7, 39-59.
- Aha, D.W. - Kibler, D., - Albert, M. K. (1991) : Instance-based learning algorithms. *Machine Learning*, 6, 37-66.
- Anguita, D. (2001): Smart Adaptive Systems: State of the Art and Future Directions of Research. European Symposium on Intelligent Technologies, Hybrid Systems and their implementation on Smart Adaptive Systems EUNITE 2001, Tenerife, Spain.
- Bauer, E. - Kohavi, R. (1999): An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36(1/2):105-139.
- Bensusan, H. - Giraud-Carrier, C. - Kennedy, C. (2000): A higher-order Approach to Meta-learning. Workshop on Meta-Learning: Building Automatic Advice Strategies for Model Selection and Method Combination, ECML'2000.

Znalosti 2003, Petr Berka

55

Literatura (2/6)

- Berka, P. (2001): Smart Adaptive Features of Machine Learning Methods. EUNITE Technical Report.
- Berka, P. (1997): Towards Knowledge Integration via Knowledge Revision. Workshop on Dynamically Changing Domains: Theory Revision and Context Dependence Issues, ECML'97.
- Berka, P. (1997b): Recognizing Reliability of Discovered Knowledge. PKDD'97, 307-314.
- Berka, P. - Ivánek, J. (1994): Automated knowledge acquisition for PROSPECTOR-like expert systems. ECML'94, 339-342.
- Berka, P. - Sláma, M. (1998): Using Neural Nets to Learn Weights of Rules for Compositional Expert Systems. IEA-98-AIE, 511-519.
- Clark, P. - Niblett, T. (1989): The CN2 induction algorithm. *Machine Learning*, 3, 261-283.
- Cortes, C. - Vapnik, V. (1995): Support vector networks. *Machine Learning*, Vol 20, 273-297.

Znalosti 2003, Petr Berka

56

Literatura (3/6)

- Dietterich, T.G. (2000): An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization. *Machine Learning*, 40(2):139-158.
- Fahlman, S.E. - Lebiere, C. (1989): The cascade correlation learning architecture. Advances in Neural Information Processing Systems.
- Fisher, D. (1987): Knowledge Acquisition via Incremental Conceptual Clustering. *Machine Learning* 2, 139-172.
- Ganascia, J.G. - Thomas, J. - Laubet, P. (1993): Integrating Models of Knowledge and Machine Learning. ECML'93, 398-401.
- Graner, N., - Sleeman, D. (1993): A Multistrategy Knowledge and Acquisition Toolbox. Workshop on Multistrategy Learning, 107-119.
- Janikow, C.Z. (1989): The AQ16 Inductive Learning Program: Some Experimental Results with AQ16 and Other Symbolic and Nonsymbolic Programs. Rep. of AI Center, George Mason University.
- Kotek, Z. - Chalupa, V. - Brůha, I. - Jelínek, J. (1980): Adaptivní a učící se systémy. SNTL, Praha

Znalosti 2003, Petr Berka

57

Literatura (4/6)

- Mahoney, J.J. - Mooney, R.J. (1994): Comparing Methods for Refining Certainty-Factor Rule-Bases. ML94.
- Matwin, S., - Kubat, M. (1996): The Role of Context in Concept Learning. Workshop on Learning in Context-Sensitive Domains, ICML'96.
- Michalski, R. (1969): On the Quasi-minimal solution of the general covering problem. FCIP'69, 125-128.
- Mitchell, T. (1997): Machine Learning. McGraw-Hill.
- Muggleton, S. (1992): Inductive Logic Programming. Academic Press.
- Ourston, D. - Mooney, R. (1994): Theory Refinement Combining Analytical and Empirical Methods. *Artificial Intelligence* 66, 311-344.
- Quinlan, J.R. (1979): Discovering rules by induction from large collections of examples. In: Expert Systems in the Micro-Electronic Age.
- Quinlan, J.R. (1986): Induction of decision trees. *Machine Learning*, 1(1), 81-106.

Znalosti 2003, Petr Berka

58

Literatura (5/6)

- Ruping, S. (2001): Incremental Learning with Support Vector Machines. Proc. ICDM 2001.
- Saitta, L. - Botta, M. - Neri, F. (1993): Multistrategy Learning and Theory Revision. *Machine Learning*, 11, 153-172.
- Shapire, R. (1999): Theoretical Views of Boosting and Applications. ALT99.
- Schlimmer, J. - Fischer, D. (1986): A Case Study of Incremental Concept Induction. IJCAI'86.
- Schlimmer, J.C. - Granger, R.H. (1986): Incremental Learning from Noisy Data. *Machine Learning* 1, 317-354.
- Torgo, L. (1992): YAILS: an Incremental Learning Program. LIACC Tech. Report, 92.1.
- Towell, G.G. - Shavlik, J. (1993): Extracting refined rules from knowledge based neural networks. *Machine Learning* 13(1), 71-102.

Znalosti 2003, Petr Berka

59

Literatura (6/6)

- Isumoto, S. - Tanaka, H. (1997): Incremental Learning Probabilistic Rules from Clinical Databases based on Rough Set Theory. *Journal of AMIA*, 4, 198-202.
- Utgoff, P.E. - Berkman, N.C. - Clouse, J.A. (1997): Decision tree induction based on efficient tree restructuring. *Machine Learning*, 29, 5-44.
- Watson, I. - Marir, F. (1994): Case-Based Reasoning: A Review. *Knowledge Engineering Review*, 9, 327-354.
- Widmer, G. - Kubat, M. (1996): Learning in the Presence of Concept Drift and Hidden Contexts, *Machine Learning*, 23(1), 69-101.
- Witten, I.H. - Frank, E. (1999): Data Mining. Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufman.
- Zadeh, L. (1994): Fuzzy Logic, Neural Networks and Soft Computing, *Communications of the ACM*, 37(3):77-84.

Znalosti 2003, Petr Berka

60