

UMELÁ INTELIGENCIA

Strojové učenie sa

Strojové učenie sa

- Simon [1983] navrhol, aby sa pod pojmom učenie sa rozumeli:

„... zmeny v systéme, ktoré sú adaptívne v tom zmysle, že umožňujú, aby systém splnil tú istú úlohu alebo úlohy z tej istej triedy úloh napodruhé efektívnejšie a účinnejšie.“

2

Strojové učenie sa – druhy

- Biffovanie
 - Samuel (1963): zapamätanie si ohodnotení z predchádzajúcej hry
- Učenie sa prispôbovaním parametrov
 - $V_1 a_1 + V_2 a_2 + \dots + V_{16} a_{16}$
 - hra dvoch programov proti sebe
- Učenie sa s makro-operátormi

3

Učiaci sa znalostný agent

- výkonná časť
 - výkonný prvok
 - snímače
 - efektoxy
- učiacia sa časť
 - učiaci sa prvok
 - hodnotiteľ konania
 - generátor problémov

4

Učenie sa s/bez učiteľa

- učenie sa s učiteľom (supervised learning, s dohľadom)
 - okamžite sú dostupné vnemy o vstupoch aj výstupoch
- učenie sa s odmenou a trestom (reinforcement learning, s posilňovaním)
 - agent dostane informáciu o hodnotení jeho akcie, ale nie o tom, aká je správna akcia
- učenie sa bez učiteľa (unsupervised learning, bez dohľadu)
 - agent nedostáva nijakú informáciu o tom, aké by mali byť správne akcie

5

Induktívne učenie sa

```
global príklady ← {}  
function VÝKONNÝ-PRVOK-S-ODRAZOM(vnem) returns akcia  
  if (vnem, a) in príklady then return a  
  else  
    h ← INDUKUJ(príklady)  
    return h(vnem)
```

```
function UČIACI-SA-PRVOK-S-ODRAZOM(vnem, akcia)  
  inputs: vnem, spätnoväzobný vnem  
         akcia, spätnoväzobná akcia  
  príklady ← príklady ∪ {(vnem, akcia)}
```

6

Učenie sa s učiteľom

Dané: Trénovacie príklady $(\mathbf{x}; f(\mathbf{x}))$, t.j. trénovacia množina

$$\{(\mathbf{x}_1, f(\mathbf{x}_1)), (\mathbf{x}_2, f(\mathbf{x}_2)), \dots, (\mathbf{x}_p, f(\mathbf{x}_p))\}$$

pre nejakú neznámu funkciu f

$$\mathbf{y} = f(\mathbf{x})$$

Nájdí:

$$f(\mathbf{x})$$

dobré priblíženie (aproximáciu) funkcie f , t.j. predpovedaj

$$\mathbf{y}' = f(\mathbf{x}')$$

kde \mathbf{x}' nie je v trénovacej množine

7

Učenie sa s učiteľom

Príklady použitia

- **rozpoznanie rukopisu**
 - \mathbf{x} : údaje o pohybe pera
 - $f(\mathbf{x})$: písmeno abecedy
- **diagnóza choroby**
 - \mathbf{x} : vlastnosti pacienta (symptómy, výsledky laboratórnych testov)
 - $f(\mathbf{x})$: choroba (alebo možno dokonca odporúčaná liečba)
- **opoznanie osoby**
 - \mathbf{x} : bitmapový obraz tváre osoby
 - $f(\mathbf{x})$: meno osoby
- **zistenie spamu**
 - \mathbf{x} : e-správa
 - $f(\mathbf{x})$: spam or nie-spam.

8

Učenie sa s učiteľom – kedy je vhodné použiť?

- **Situácie, keď nemáme ľudského experta**
- **Situácie, keď ľudia vedia riešiť problém, ale nevedia opísať, ako to robia**
 - \mathbf{x} : bitmapový obraz znaku napísaného rukou
 - $f(\mathbf{x})$: ascii kód toho znaku
- **Situácie, keď sa požadovaná funkcia často mení**
 - \mathbf{x} : opis cien akcií a obchodov za posledných 10 dní
 - $f(\mathbf{x})$: odporúčané obchody
- **Situácie, keď každý zákazník potrebuje personalizovanú funkciu f**
 - \mathbf{x} : prichádzajúca e-správa
 - $f(\mathbf{x})$: dôležitosť správy (skóre udávajúce naliehavosť, s akou správou zákazníkovi prezentovať alebo naopak zrušiť bez prezentovania)

9

klasifikovanie

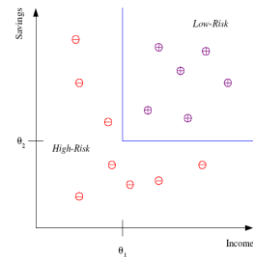
- príklad: hodnotenie úverov
- Rozlišovanie medzi **nízko-rizikovými** a **vysoko-rizikovými** zákazníkmi na základe ich príjmu a úspor

Diskriminant:

IF $\text{príjem} > \theta_1$ A $\text{úspory} > \theta_2$

THEN **nízke-riziko**

ELSE **vysoké-riziko**



10

opoznanie osoby (tváre)

Trénovacie príklady obrazu tváre osoby



Testovacie obrazy

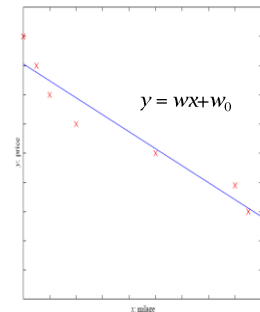


11

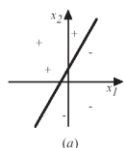
AT&T Laboratories, Cambridge UK
<http://www.ukresearch.att.com/face/database.html>

Regresia

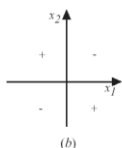
- príklad: cena jazdeného auta
- \mathbf{x} : atribúty auta
- y : cena
- $y = g(\mathbf{x} | \theta)$
- $g(\cdot)$ model,
- θ parametre



Lineárna regresia



lineárne separovateľný



nie je lineárne separovateľný

13

Rozhodovacie stromy

Učenie sa pomocou rozhodovacích stromov je všeobecne najvhodnejšia metóda na riešenie problémov s nasledujúcou charakteristikou:

- Inštancie sú reprezentované dvojicou atribút - hodnota.
- Cieľová funkcia má diskkrétne výstupné hodnoty.
- Problémy vyžadujúce disjunktívny opis
- Rozhodovacie stromy prirodzene reprezentujú disjunktívne výrazy.
- Údaje na tréningovanie obsahujú chyby
- Tréningové údaje môžu obsahovať chýbajúce hodnoty atribútov

Učiace funkcie sú buď reprezentované rozhodovacími stromami alebo reprezentované množinou if - then pravidiel na zlepšenie čitateľnosti.

14

Príklad - Rozhodovacie stromy

Dávid je manažérom golfového klubu. Sú dni, keď chce hrať golf každý a personál ihriska je preťažený, inokedy však golf nehra nik a personál klubu má príliš veľa voľného času.

Dávidovým cieľom je optimalizovať dostupnosť personálu podľa predpovede, koľko ľudí bude v ten ktorý deň hrať golf. Počas 14 dní si zaznamenával, aké bolo počasie, vlhkosť vzduchu, rýchlosť vetra a či ľudia hrali alebo nehrali golf.

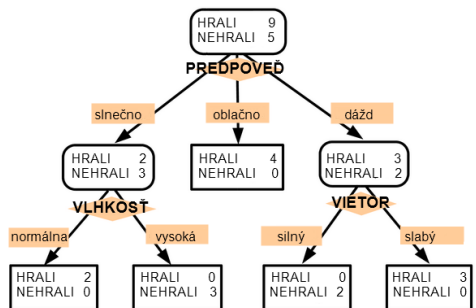
15

Príklad - Rozhodovacie stromy

DEŇ	PREDPOVEĎ	VLHKOSŤ	VIETOR	HRALI
1	Slnečno	Vysoká	Slabý	Nie
2	Slnečno	Vysoká	Silný	Nie
3	Oblačno	Vysoká	Slabý	Áno
4	Dážď	Vysoká	Slabý	Áno
5	Dážď	Normálna	Slabý	Áno
6	Dážď	Normálna	Silný	Nie
7	Oblačno	Normálna	Silný	Áno
8	Slnečno	Vysoká	Slabý	Nie
9	Slnečno	Normálna	Slabý	Áno
10	Dážď	Normálna	Slabý	Áno
11	Slnečno	Normálna	Silný	Áno
12	Oblačno	Vysoká	Silný	Áno
13	Oblačno	Normálna	Slabý	Áno
14	Dážď	Vysoká	Silný	Nie

16

Príklad - Rozhodovacie stromy



17

Príklad - Rozhodovacie stromy

IF (Predpoveď = slnečno) ∧ (Vlhkosť = normálna)

THEN Hrali = **ÁNO**

IF (Predpoveď = slnečno) ∧ (Vlhkosť = vysoká)

THEN Hrali = **NIE**

IF (Predpoveď = oblačno) THEN hrali = **ÁNO**

IF (Predpoveď = dážď) ∧ (Vietor = silný)

THEN Hrali = **NIE**

....

18

Induktívne vytváranie rozhodovacích stromov

```

function UČENIE-SA-ROZHODOVACIEHO-STROMU(priklady, atribúty, štandard)
returns rozhodovací strom
inputs: priklady, množina príkladov
        atribúty, množina atribútov
        štandard, štandardná hodnota cieľovej hypotézy

if priklady sú prázdne then return štandard
else if všetky priklady majú rovnakú klasifikáciu then return klasifikácia
else if atribúty sú prázdne then return HODNOTA-VÁČŠINY(priklady)
else
    najlepší ← VYBER-ATRIBÚT(atribúty, priklady)
    strom ← nový rozhodovací strom s koreňom s testom na najlepší
    for each hodnota vi atribútu najlepší do
        priklady ← {prvky z prikladov s hodnotou atribútu najlepší = vi}
        podstrom ← UČENIE-SA-ROZHODOVACIEHO-STROMU
                     (priklady, atribúty - najlepší, HODNOTA-VÁČŠINY(štandard))
        pridaj hranu do stromu s ohodnotením vi a podstromom podstrom
    end
return strom

```

19

Posudzovanie kvality algoritmu učenia sa

- klasifikácia príkladu hypotézou
- testovacia množina
 1. Nazhromaždiť veľkú množinu príkladov.
 2. Rozdeliť ich do dvoch disjunktných množín: tréningovej a testovacej.
 3. Použiť učiaci sa algoritmus s tréningovou množinou ako množinou príkladov na vytvorenie hypotézy H .
 4. Zmerať podiel príkladov v testovacej množine, ktoré sa správne klasifikovali podľa hypotézy H .
 5. Opakovať kroky 1 až 4 pre rôzne veľkosti tréningových množín a pre rôzne náhodne zvolené testovacie množiny pre každú veľkosť.

20

Učenie sa všeobecných logických opisov

- priestor hypotéz

$$H_1 \vee H_2 \vee \dots \vee H_n$$
- zlučiteľnosť hypotéz
 - falošný negatívny príklad
 - falošný pozitívny príklad

21

Učenie sa hľadaním pomocou najlepšej súčasnej hypotézy

```

function UČENIE-SA-SÚČASNEJ-NAJLEPŠEJ-HYPOTÉZY(priklady)
returns hypotéza
inputs: priklady, množina príkladov

H ← ľubovoľná hypotéza zlučiteľná s prvým príkladom v prikladoch
for each zvyšujúci príklad p v prikladoch do
    if p je falošný pozitívny pre H then
        H ← choose špecializácia hypotézy H zlučiteľná s príkladmi
    else if p je falošný negatívny pre H then
        H ← choose zovšeobecnenie hypotézy H zlučiteľné s príkladmi
    if nedá sa nájsť zlučiteľné zovšeobecnenie ani špecializácia then fail
end
return H

```

22

Učenie sa hľadaním v priestore verzií

```

function UČENIE-SA-V-PRÍESTORE-VERZIÍ(priklady)
returns priestor verzií
inputs: priklady, množina príkladov
local variables: V, priestor verzií: množina všetkých hypotéz

V ← množina všetkých hypotéz
for each príklad p v prikladoch do
    if V nie je prázdny then
        V ← OBNOVA-PRÍESTORU-VERZIÍ(V, p)
    end
return V

```

```

function OBNOVA-PRÍESTORU-VERZIÍ(V, p)
returns obnovený priestor verzií

```

$V \leftarrow \{h \in V : h \text{ je zlučiteľná s } p\}$

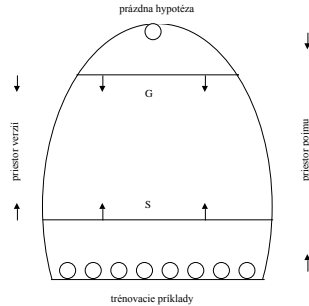
23

Učenie sa hľadaním v priestore verzií

- priestor verzií
 - G-množina, S-množina
- obnova priestoru verzií
 - Falošný pozitívny príklad (FPP) pre S_j ; S_j je príliš všeobecná, vylúčime ju z S-množiny
 - FNP pre S_j ; S_j je príliš zvláštna, nahradíme ju všetkými jej možnými bezprostrednými zovšeobecneniami
 - FPP pre G_j ; G_j je príliš všeobecná, nahradíme ju všetkými jej možnými bezprostrednými špecializáciami
 - FNP pre G_j ; G_j je príliš zvláštna, vylúčime ju z G-množiny

24

Priestor pojmu a priestor verzii



25

Algoritmus odstraňovania kandidátov

Výstupom je opis pojmu, ktorý je zlučiteľný so všetkými pozitívnymi príkladmi a s nijakým negatívnym príkladom.

1. Inicializuj G tak, aby obsahovala práve jeden prvok: prázdny opis
2. Inicializuj S tak, aby obsahovala práve jeden prvok: prvý pozitívny príklad.
3. Vezmi ďalší trénovací príklad.
Ak je to pozitívny príklad,
tak zovšeobecni prvky v S čo najmenej tak, aby boli zlučiteľné s novým trénovacím príkladom.
Ak je to negatívny príklad,
tak špecializuj prvky v G čo najmenej tak, aby nový trénovací príklad už nebol zlučiteľný s hocijakým prvkom v G
4. **Ak** sú S aj G obe jednoprvkové množiny,
tak ak sú rovnaké, **tak** vráť ten prvok a zastav,
inak oznám, že množina trénovacích príkladov bola nezlučiteľná a zastav,
inak pokračuj krokom 3.

26

Príklad – Algoritmus odstraňovania kandidátov

Uvažujme problémovú oblasť historických stavebných pamiatok. Predpokladajme ďalej, že v každej položke sa môžu vyskytovať len tieto hodnoty:

krajina \in {Slovensko, Francúzsko, Škótsko, Česko}
 druh \in {kostol, hrad, zámok, palác}
 sloh \in {románsky, gotika, renesancia, barok, neoklasicizmus}
 stav \in {zachovalý, poškodený, zrúcanina}

Ako prebehne učenie sa pojmu „slovenská gotická pamiatka“ ??

pamiatka_slovenska_goticka
 krajina : Slovensko
 druh : x1
 sloh : gotika
 stav : x2

27

Príklad – Algoritmus odstraňovania kandidátov

Predpokladajme, že v trénovacej množine budú tieto príklady:

(1:pozitívny)		(2:negatívny)	
krajina:	Slovensko	krajina:	Slovensko
druh:	hrad	druh:	kostol
sloh:	gotika	sloh:	renesancia
stav:	poškodený	stav:	zachovalý
(3:pozitívny)		(4:negatívny)	
krajina:	Slovensko	krajina:	Francúzsko
druh:	kostol	druh:	kostol
sloh:	gotika	sloh:	gotika
stav:	poškodený	stav:	zachovalý
(5:pozitívny)			
krajina:	Slovensko		
druh:	kostol		
sloh:	gotika		
stav:	zachovalý		

28

Príklad – Algoritmus odstraňovania kandidátov

- Na začiatku budú G aj S jednoprvkové množiny opisov
 $G = \{(x1, x2, x3, x4)\}$
 $S = \{(Slovensko, hrad, gotika, poškodený)\}$
- Druhý príklad je negatívny. Množinu G treba špecializovať tak, aby tento negatívny príklad už nepatril do priestoru verzii (premenná sa nahradí konštantou).
 $G = \{(x1, hrad, x3, x4), (x1, x2, gotika, x4), (x1, x2, x3, poškodený)\}$
 Množina S sa spracovaním druhého príkladu nezmení.
- Tretí príklad je opäť pozitívny. Najprv sa z G odstráni všetky opisy, ktoré sú s týmto príkladom nezlučiteľné. S sa zovšeobecniť tak, aby zahŕňala aj tento príklad (nahradením konštanty premennou).
 $G = \{(x1, x2, gotika, x4), (x1, x2, x3, poškodený)\}$
 $S = \{(Slovensko, x2, gotika, poškodený)\}$

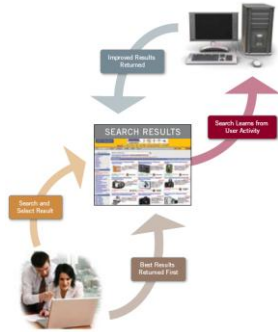
29

Príklad – Algoritmus odstraňovania kandidátov

- Ďalší príklad je negatívny. Je to pamiatka, ktorá leží vo Francúzsku. Množina S sa nezmení. Množinu G treba špecializovať, aby nepokrýla tento príklad:
 $G = \{(Slovensko, x2, gotika, x4), (Slovensko, x2, x3, poškodený)\}$
 $S = \{(Slovensko, x2, gotika, poškodený)\}$
- Posledný príklad je pozitívny. Aby s ním zostala množina G zlučiteľná, treba z nej odstrániť druhý prvok a množinu S zovšeobecniť tak, aby zahŕňala aj nový príklad:
 $G = \{(Slovensko, x2, gotika, x4)\}$
 $S = \{(Slovensko, x2, gotika, x4)\}$
- Teraz sú obe množiny S aj G jednoprvkové. Sú rovnaké, takže sa podarilo nájsť opis pojmu, ktorý bolo cieľom naučiť sa.

30

Príklad – Učenie sa pre hľadanie na webe



31

Príklad – Učenie sa pre hľadanie na webe

- Ako získať dáta pre zdokonalenie vyhodnocovacej funkcie ?
 - Explicitná vs. implicitná spätná väzba
 - Absolútna vs. relatívna spätná väzba
 - Štúdia sledovania pohybu očí

32

Príklad – Učenie sa pre hľadanie na webe

Explicitná spätná väzba

- Obťažuje používateľ
- Nízka odozva
- > málo reprezentujúca



Implicitná spätná väzba

- Dotazy, kliknutia, čas strávený na stránke, skrolovanie, atď
- Neobťažuje používateľ
- Horšia interpretácia

33

Príklad – Učenie sa pre hľadanie na webe

Relatívna spätná väzba:
Kliknutie znamená uprednostnenie odkazu pred ostatnými zobrazenými.

Absolútna spätná väzba:
Odkazy, na ktoré používateľ klikol sú relevantné k vyhľadávaciemu dopytu.

(3 < 2),
(7 < 2),
(7 < 4),
(7 < 5),
(7 < 6)

1. Kernel Machines
<http://www.first.gmd.de/>
2. Support Vector Machine
<http://jbalzar.freemove.com/>
3. SVM-Light Support Vector Machine
<http://svt.gmd.de/~thoresen/svm-light/>
4. An Introduction to Support Vector Machines
<http://www.support-vector.net/>
5. Support Vector Machine and Kernel ... References
<http://www.research.bell-labs.com/SVMrefs.html>
6. Archives of SUPPORT-VECTOR-MACHINES ...
<http://www.jiscmail.ac.uk/lists/SUPPORT...>
7. Lucient Technologies SVM demo applet
<http://svm.research.bell-labs.com/SVT-SVMsvt.html>
8. Royal Holloway Support Vector Machine
<http://svm.dcs.rhmc.ac.uk>

34

Je implicitná spätná väzba spoľahlivá ?

- Podľa čoho sa používateľ rozhoduje, kde kliknúť ?
- Koľko výsledkov si používateľ prezrie predtým, ako na nejaký klikne ?
 - Prezerá používateľ výsledky vyhľadávania odhora nadol ?
 - Prezrie si používateľ všetky výsledky vyhľadávania nad odkazom, na ktorý klikol ?
 - Sú pre používateľa zaujímavé aj výsledky vyhľadávania zobrazené pod odkazom, na ktorý klikol ?

Ako súvisí kliknutie na odkaz s jeho relevantnosťou ?

- Absolútna spätná väzba:
 - Sú odkazy, na ktoré používateľ klikol relevantné ?
 - Sú odkazy, na ktoré používateľ neklikol nerelevantné ?
- Relatívna spätná väzba:
 - Sú odkazy, na ktoré používateľ klikol viac relevantné ako odkazy, na ktoré používateľ neklikol ?

1. Kernel Machines
<http://www.kernel-machines.org/>
2. Support Vector Machine
<http://jbalzar.freemove.com/>
3. SVM-Light Support Vector Machine
<http://svt.gmd.de/~thoresen/svm-light/>
4. An Introduction to SVMs
<http://www.support-vector.net/>
5. Support Vector Machine and ...
<http://www.research.bell-labs.com/SVMrefs.html>
6. Archives of SUPPORT-VECTOR-MACHINES ...
<http://www.jiscmail.ac.uk/lists/SUPPORT...>
7. Lucient Technologies SVM demo applet
<http://svm.research.bell-labs.com/SVT-SVMsvt.html>
8. Royal Holloway SVM
<http://svm.dcs.rhmc.ac.uk>
9. SVM World
<http://www.svmworld.com>
10. Transilvania FIRST SVM page
<http://svm.first.gmd.de>

35

Príklad – Učenie sa pre hľadanie na webe



Štúdia sledovania pohybu očí

- Fixácia:
 - ~200 – 300ms, získanie informácie
- Kmitanie oka:
 - extrémne rýchle pohyby medzi jednotlivými fixáciami
- Rozšírenie zreničiek:
 - veľkosť zreničky súvisí so záujmom používateľa o prezeranú tému



36

Príklad – Učenie sa pre hľadanie na webe

- Koľko výsledkov vyhľadávania používateľ prezrie?



37

Príklad – Učenie sa pre hľadanie na webe

- Používatelia najčastejšie prezrú 2 výsledky vyhľadávania.
- Používatelia zvyčajne prezerajú výsledky vyhľadávania odhora nadol.
- Používatelia najčastejšie kliknú na prvý odkaz na stránke s výsledkami vyhľadávania.
- Používatelia sa zvyčajne nepozerajú na odkazy, ktoré sú zobrazené nižšie ako ten, na ktorý klikli.

38