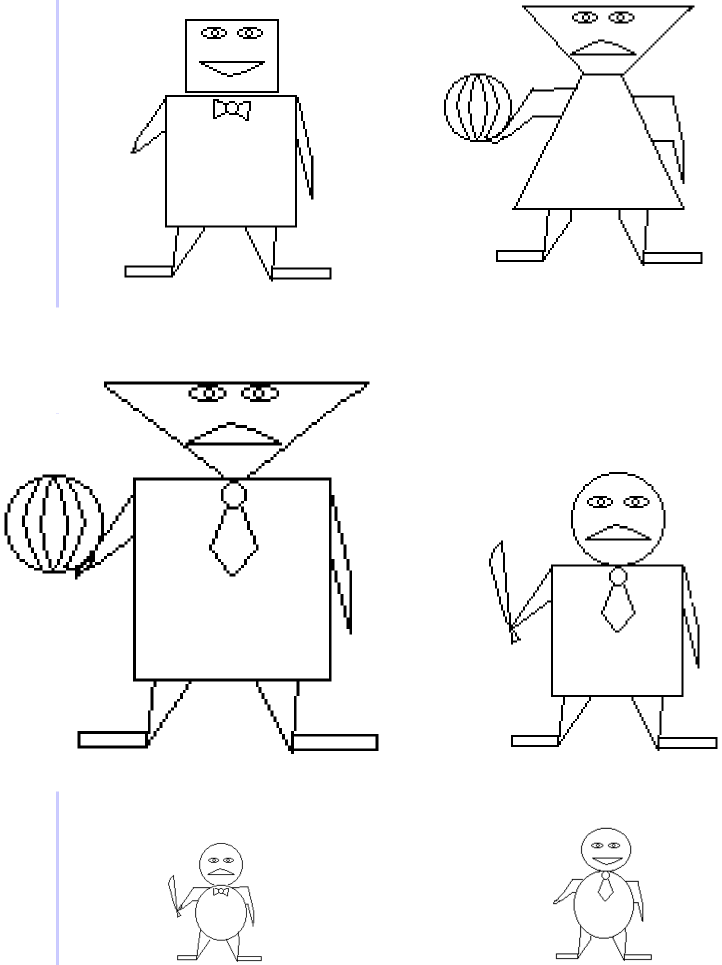
Rozhodovací stromy a jejich konstrukce z dat

Nature Inspired
Technologies Group

Příklad „počítačová hra“.

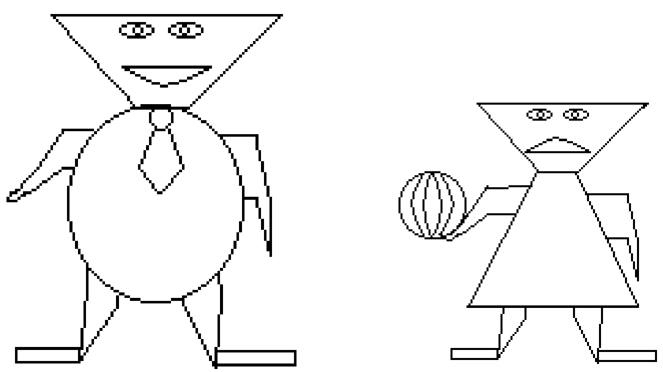
Můžeme počítač naučit rozlišovat přátelské a nepřátelské roboty?

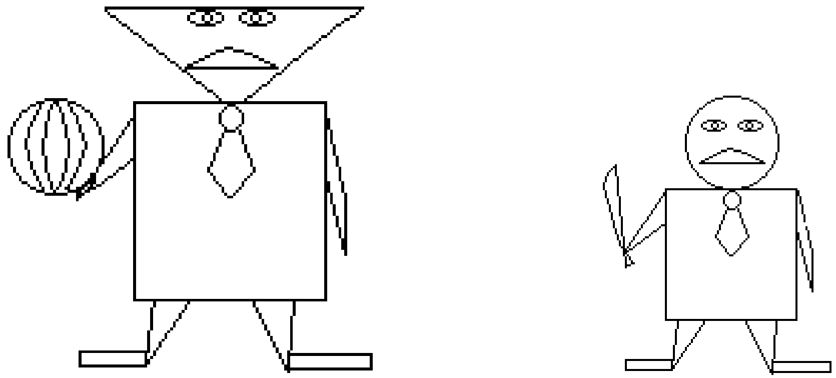


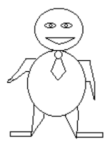
Učení s učitelem:
u některých už víme, jakou mají povahu (**klasifikace**)

Neparametrická úloha:
Nic nevíme o pravděpodobnostní distribuci jednotlivých objektů

Příklad „počítačová hra“ 1. Můžeme se naučit roboty rozlišit na základě krátké zkušenosti?

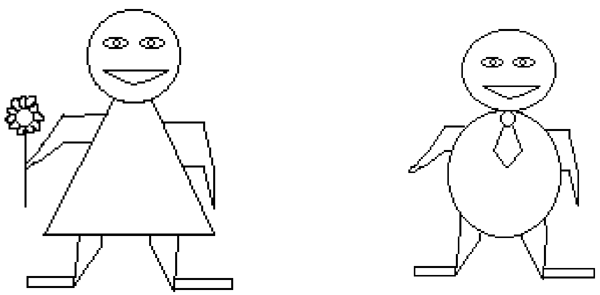


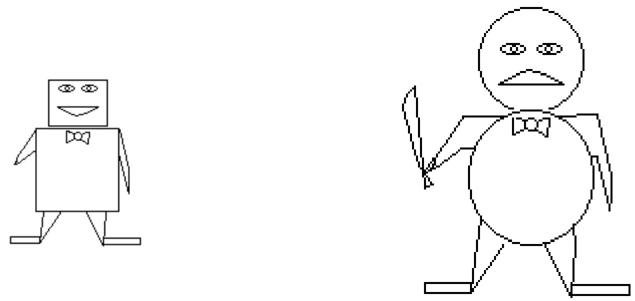




přátelští

nepřátelští

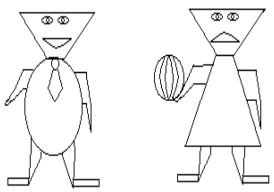


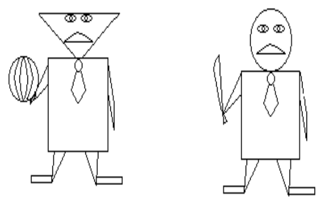


Reprezentace úlohy pomocí atributů

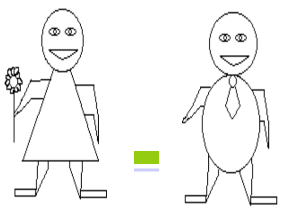
Klasifikace	Usmíva_se	kravata	tělo	hlava	v_ruce
přítel	ano	ano	kruh	3úhelník	nic
přítel	ne	ne	3úhelník	3úhelník	balon
nepřítel	ne	ano	čtverec	3úhelník	balón
nepřítel	ne	ano	čtverec	kruh	meč
přítel	ano	ne	3úhelník	kruh	květ
přítel	ano	ano	kruh	kruh	nic
nepřítel	ano	ne	čtverec	čtverec	nic
nepřítel	ne	ne	kruh	kruh	meč

Můžeme navrhnout odpovídající klasifikační algoritmus ?

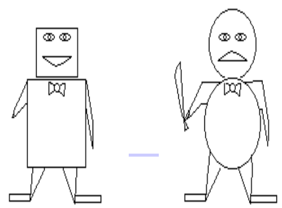




přátelští



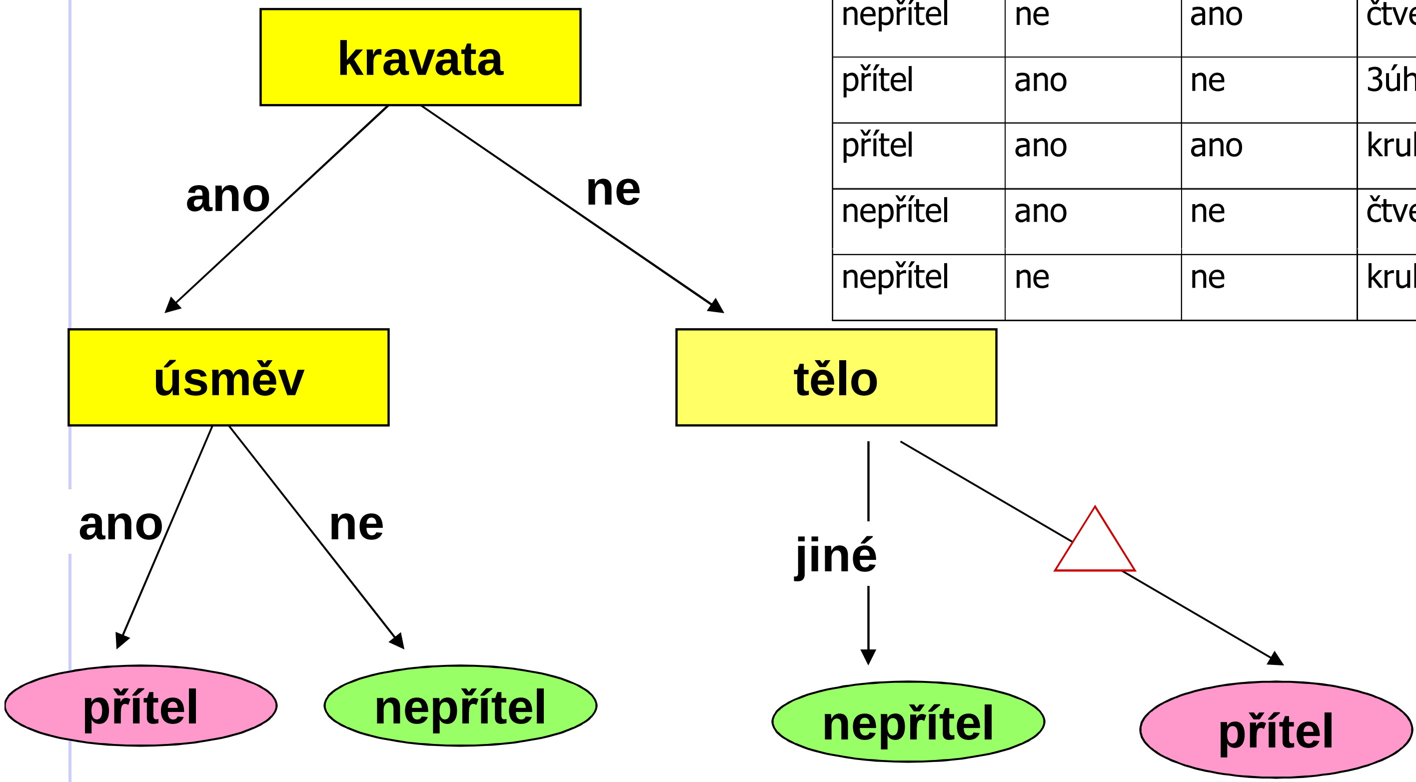
nepřátelští



Vyhledávání v tabulce, …

Rozhodovací strom 1 pro danou množinu příkladů

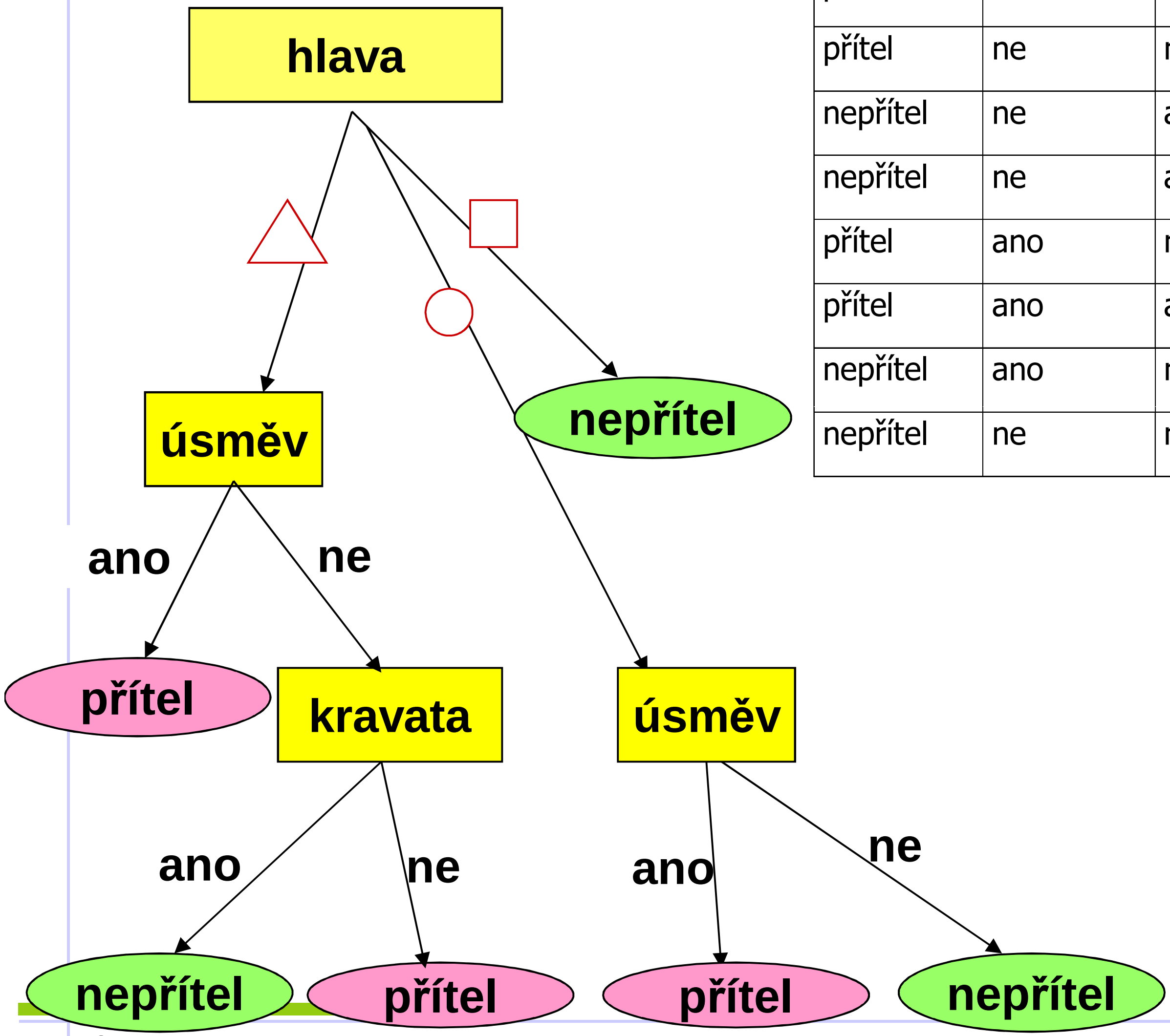
Klasifikace	Usmíva_se	kravata	tělo	hlava	v_ruce
přítel	ano	ano	kruh	3úhelník	nic
přítel	ne	ne	3úhelník	3úhelník	balon
nepřítel	ne	ano	čtverec	3úhelník	balón
nepřítel	ne	ano	čtverec	kruh	meč
přítel	ano	ne	3úhelník	kruh	květ
přítel	ano	ano	kruh	kruh	nic
nepřítel	ano	ne	čtverec	čtverec	nic
nepřítel	ne	ne	kruh	kruh	meč



Nature Inspired Technologies Group

Rozhodovací strom 2 pro tutéž množinu příkladů

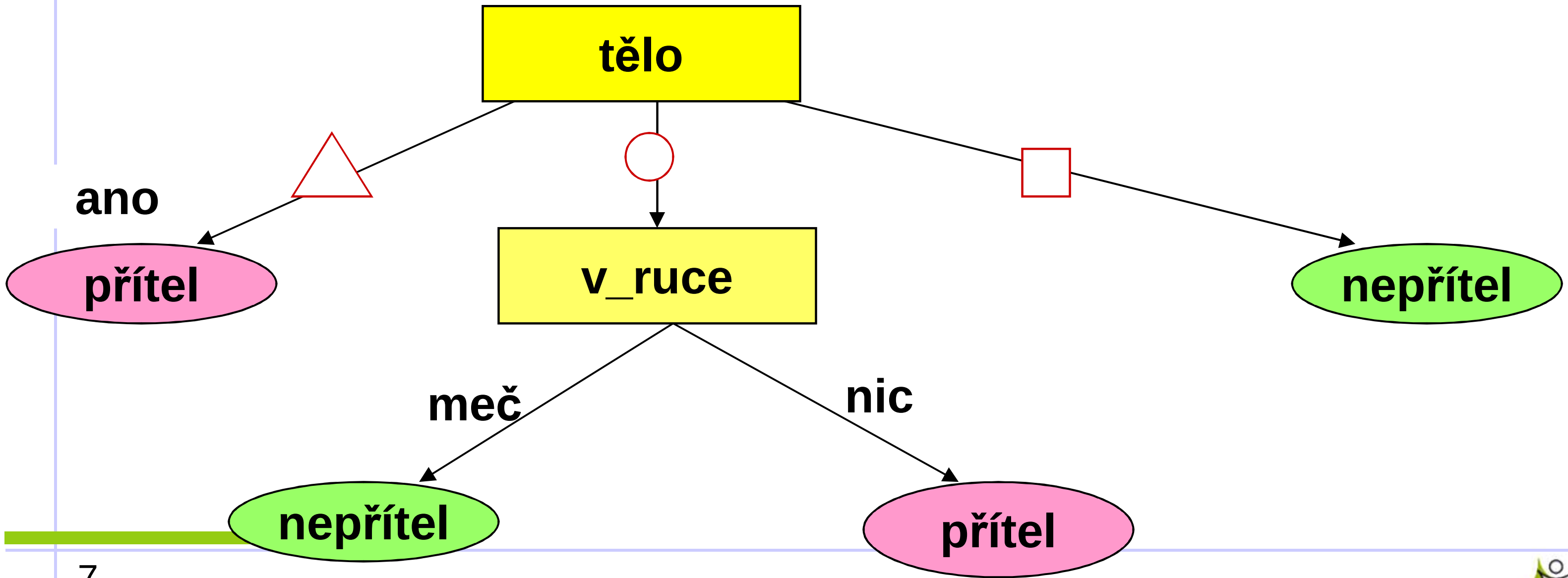
Klasifikace	Usmíva_se	kravata	tělo	hlava	v_ruce
přítel	ano	ano	kruh	3úhelník	nic
přítel	ne	ne	3úhelník	3úhelník	balon
nepřítel	ne	ano	čtverec	3úhelník	balón
nepřítel	ne	ano	čtverec	kruh	meč
přítel	ano	ne	3úhelník	kruh	květ
přítel	ano	ano	kruh	kruh	nic
nepřítel	ano	ne	čtverec	čtverec	nic
nepřítel	ne	ne	kruh	kruh	meč



Rozhodovací strom 2 pro tutéž množinu příkladů

Klasifikace	Usmíva_se	kravata	tělo	hlava	v_ruce
přítel	ano	ano	kruh	3úhelník	nic
přítel	ne	ne	3úhelník	3úhelník	balon
nepřítel	ne	ano	čtverec	3úhelník	balón
nepřítel	ne	ano	čtverec	kruh	meč
přítel	ano	ne	3úhelník	kruh	květ
přítel	ano	ano	kruh	kruh	nic
nepřítel	ano	ne	čtverec	čtverec	nic
nepřítel	ne	ne	kruh	kruh	meč

Který strom je lepší
a jak jej najdeme?



Klasifikace	**úsměv**	**kravata**	**tělo**	**hlava**	**v_ruce**
přítel	ano	ano	kruh	3úhelník	nic
přítel	ne	ne	3úhelník	3úhelník	balon
nepřítel	ne	ano	čtverec	3úhelník	balón
nepřítel	ne	ano	čtverec	kruh	meč
přítel	ano	ne	3úhelník	kruh	květ
přítel	ano	ano	kruh	kruh	nic
nepřítel	ano	ne	čtverec	čtverec	nic
nepřítel	ne	ne	kruh	kruh	meč

Souhrn	**úsměv**	**Kravata**	**tělo=3úh.**	**hlava=3úh**	**v_r.=nic**
významu atributů	Ano:3P,1N Ne: 1P,3N	Ano:2P,2N Ne: 2P,2N	Ano:2P,0N Ne: 2P,4N	Ano:2P,1N Ne:2P,3N	Ano:2P,1N Ne: 2P,3N

Indukce rozhodovacího stromu z trénovací množiny

dáno: **S** ... trénovací množina (množina klasifikovaných příkladů)

1. Nalezni "nejlepší" atribut $\mathbf{at}_0$ (t.j. atribut, jehož hodnoty nejlépe diskriminují mezi pozitivní a neg. příklady) pro **S** a tím ohodnoť kořen vytvářeného stromu.

2. Rozděl množinu **S** na podmnožiny $\mathbf{S_1, S_2, \ldots, S_n}$ podle hodnot atributu $\mathbf{at}_0$ a pro každou množinu příkladů $\mathbf{S_i}$ vytvoř nový uzel jako následníka právě zpracovávaného uzlu (kořenu)

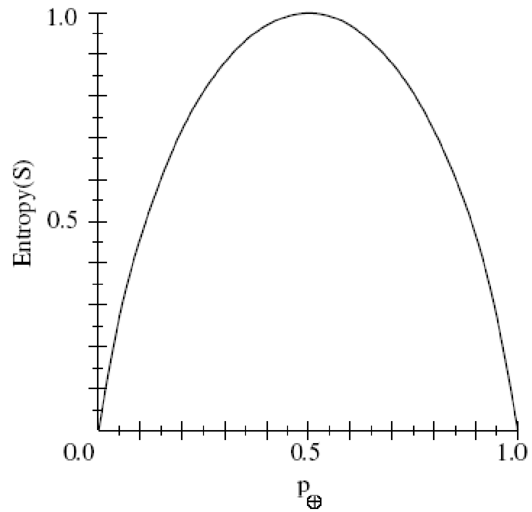
3. Pro každý nově vzniklý uzel s přiřazenou podmnožinou $\mathbf{S_i}$ proveď:

- **Jestliže** všechny příklady v $\mathbf{S_i}$ mají tutéž klasifikaci (všechny jsou pozitivní nebo všechny jsou negativní),
- **pak** uzel ohodnocený $\mathbf{S_i}$ je prohlášen za list vytvářeného rozhodovacího stromu (a tedy se už dále nevětví),
- **jinak** jdi na bod **1** s tím, že **S** : = $\mathbf{S_i}$.

Entropie množiny S vzhledem k dané klasifikaci

- Posuzuje „různorodost“ klasifikace prvků z množiny S

- Nechť klasifikaci představuje atribut y, který má jen 2 hodnoty {0,1}. Pak označme $S^0 = \{z \in S : z_y = 0\}$ a
  $S^1 = \{z \in S : z_y = 1\}$



Entropy(S) = $- |S^0|/|S| * \log_2 |S^0|/|S| - |S^1|/|S| * \log_2 |S^1|/|S|$,

kde $|A|$ označuje mohutnost množiny A

- Je-li $S^0 = \emptyset$, pak Entropy (S)=0 …
- Je-li $|S^0| = |S^1|$, pak Entropy (S)=1
- Je-li $|S| = 1$, pak Entropy (S)= 0

Volba nejlepšího atributu pro množinu příkladů S vzhledem k dané klasifikaci

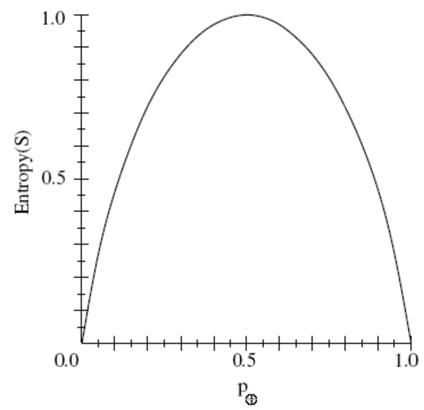
Kriterium minimální entropie rozkladu (KMER)

- Nechť ***at*** je pevně zvolený atribut, který může nabývat hodnot v_1 až v_n.
- Označme $S_i = \{z \in S : z_{at} = v_i\}$ podmnožinu S, která obsahuje právě ty objekty, které v atributu ***at*** maji hodnotu v_i
- Vážená entropie $E(S,at)$ rozkladu S podle hodnot atributu ***at*** charakterizuje „čistotu" klasifikace v jednotlivých složkách rozkladu S a je definována $E(S,at) = \sum^{n}_{i=1} |S_i|/|S| * E(S_i)$

KMER vypočte $E(S,at)$ pro všechny atributy ***at*** a jako nejlepší atribut **at^0 zvolí ten z nich, pro který je hodnota $E(S,at^0)$ nejmenší**

Základní algoritmus ID3

❖Realizuje prohledávání prostoru všech stromů, které lze zkonstruovat v jazyku trénovacích dat :



- shora dolů
- s použitím hladové strategie

❖Volba atributu pro větvení na základě charakterizace „(ne)homogenity nově vzniklého pokrytí" (používají se různé míry), např.:

- **Kriterium minimalní entropie rozkladu**
- **Informační zisk** (gain) odhaduje předpokládané snížení entropie pro pokrytí vzniklé použitím hodnot odpovídajícího atributu

$$Gain(S,A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Nebezpečí použití kriteria KMER

Co se stane, pokud některý atribut má hodně „skoro“ unikátních hodnot, které takřka jednoznačně charakterizují každý trénovací příklad? Například pro rodné číslo je $|S_i| = 1$ a tedy

$$E(S_i) = 0 \text{ a } E(S, rodne\_cislo) = 0$$

Tento agrument je tedy kriteriem KMER vybrán jako nejlepší !

Je takový atribut opravdu užitečný pro testovací data?

Nature Inspired Technologies Group

Nebezpečí použití kriteria KMER

Co se stane, pokud některý atribut má hodně „skoro“ unikátních hodnot, které takřka jednoznačně charakterizují každý trénovací příklad? Například pro rodné číslo je $|S_i| = 1$ a tedy

$$E(S_i) = 0 \text{ a } E(S, rodne\_cislo) = 0$$

Tento agrument je tedy kriteriem KMER vybrán jako nejlepší !

Je takový atribut opravdu užitečný pro testovací data?

Ne, má malou generalizační schopnost!

Nebylo by vhodné takovou situaci nějak „penalizovat“? JAK? Zde (pro KMER = 0) nepomůže multiplikatiční koeficient! Raději využijeme hodnotu doplňkovou, kterým je **kriterium zisku**:

$$Gain\ E(S,at) = E(S) - E(S,at) = E(S) - \sum^{n}_{i=1} |S_i|/|S| * E(S_i)$$

Jak charakterizovat rozklad množiny *S* na c disjunktních podmnožin S_i podle všech hodnot uvažovaného atributu ***A*** **?**

$$SplitInformation(S, A) \equiv - \sum_{i=1}^{c} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

SplitInformation odpovídá entropii rozdělení ***S*** podle všech hodnot atributu ***A*** **.** Např. je-li $|S_i| = 1$ pro všechna $i < (c+1)$, je jeho hodnota rovna $(\log_2 c)$. Používá se pro výpočet *GainRatio,* který penalizuje atributy s příliš mnoha hodnotami:

$$\boxed{GainRatio(S, A) \equiv \frac{Gain(S, A)}{SplitInformation(S, A)}}$$

Volba atributů a další „speciální situace“

- Reálné hodnoty.

→ používá se **diskretizace**

- Různé ceny pro získání hodnoty atributu

Volba atributů a „speciální situace" 1

- Reálné hodnoty → používá se **diskretizace**
- **JAK SE VOLÍ VHODNÉ MEZNÍ HODNOTY?**

Vhodné řešení (Fayyad 91): Uspořádejte příklady podle velikosti zpracovávaného atributu a zvolte jako kandidátní mezní hodnoty ty, které leží v intervalu, kde se mění klasifikace. Hodnota, která maximalizuje ***Gain***, je nutně jednou z nich.

Temperature:	40	48	60	72	80	90
PlayTennis:	No	No	Yes	Yes	Yes	No

Volba atributů a „speciální situace“ 2

❖ Různé ceny pro získání hodnoty atributu.

❖ Určíme-li cenu *Cost(A)* v intervalu <0,1>, pak použijeme změněné kriterium, např.

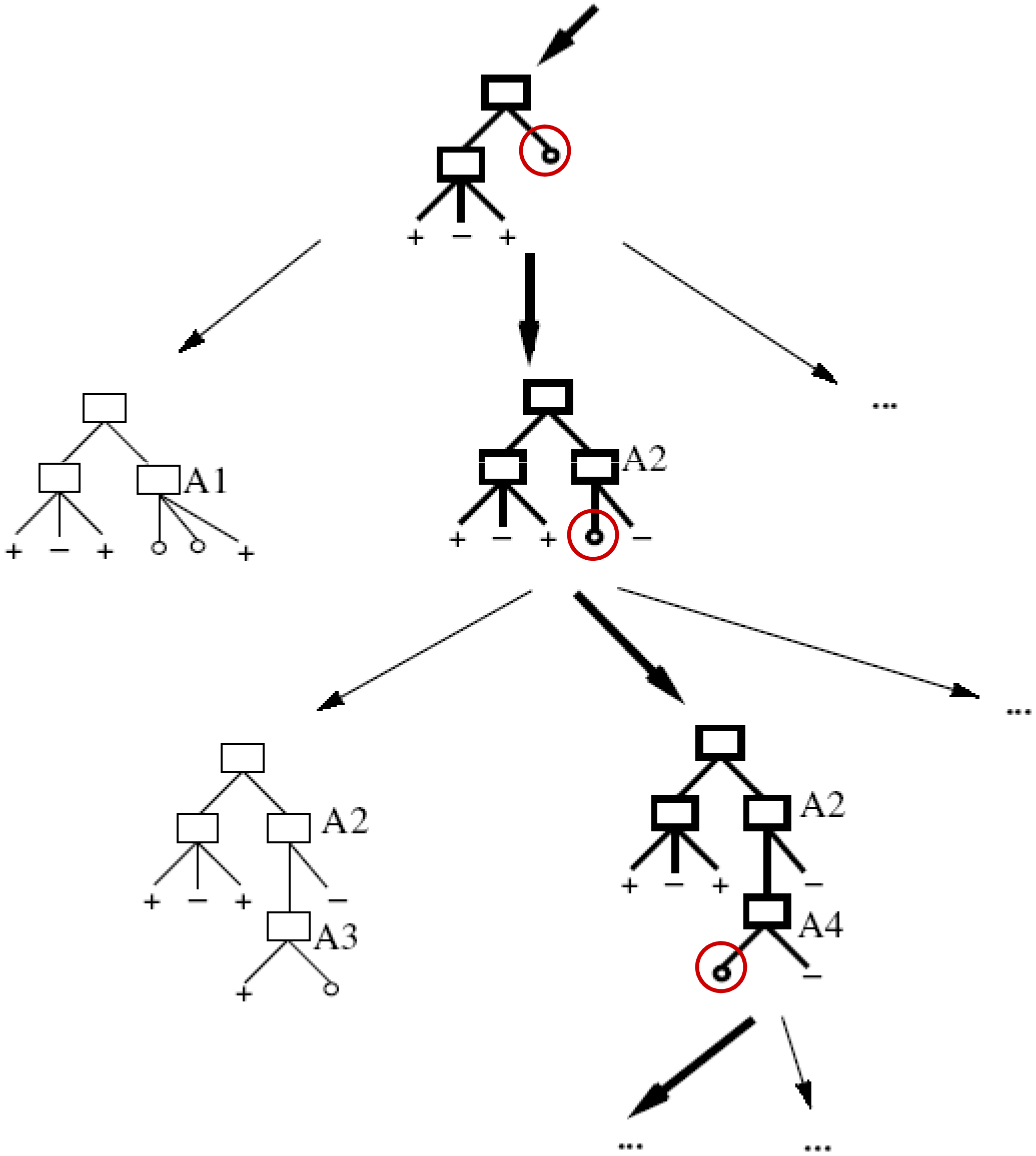
- Tan and Schlimmer (1990)

$$\frac{Gain^2(S, A)}{Cost(A)}.$$

- Nunez (1988)

$$\frac{2^{Gain(S,A)} - 1}{(Cost(A) + 1)^w}$$

Postup prohledávání



Vlastnosti ID3: důsledky postupu prohledávání

- Pro klasifikační úlohu s diskrétními atributy je prohledávaný **prostor hypotéz úplný** (tj. je schopný reprezentovat libovolnou možnou cílovou funkci) --> existuje mnoho hypotéz konzistentních s daty!
- Aktuální **množina hypotéz je vždy jednoprvková** (hlado-vá volba následníka), nelze jej tedy použít pro odpověď na dotaz „kolik je alternativních stromů konzistentních s daty?"
- Nepoužívá zpětný chod --> **možnost uvíznutí v lokálním optimu**
- **Rozhoduje se na základě všech příkladů** (nikoliv inkrementálně) --> metoda není příliš ovlivněna šumem

Kdy je vhodné použít algoritmy pro konstrukci rozhodovacího stromu?

- **Cílová funkce má diskrétní hodnoty** (jedná se o **klasifikační problém**)
- **Instance trénovacích dat mají jednotný formát** popisující hodnoty atributů
- **Trénovací data mohou**
  - **být zašuměná**
  - **Obsahovat chybějící hodnoty**
- **Je potřeba reprezentovat disjunkci podmínek (pravidla)**

Proč dáváme přednost jednoduchým hypotézám?

Argument : Jednoduchých hypotéz je výrazně méně než složitých. Proto, pokud některé z jednoduchých h. data odpovídají, pak asi nejde o „náhodný jev"

Occamova břitva :
Nejlepší hypotéza je ta nejjednodušší, která odpovídá datům.

Související problémy:
- proč zrovna **tato** malá množina?
- pozor na použitý jazyk!

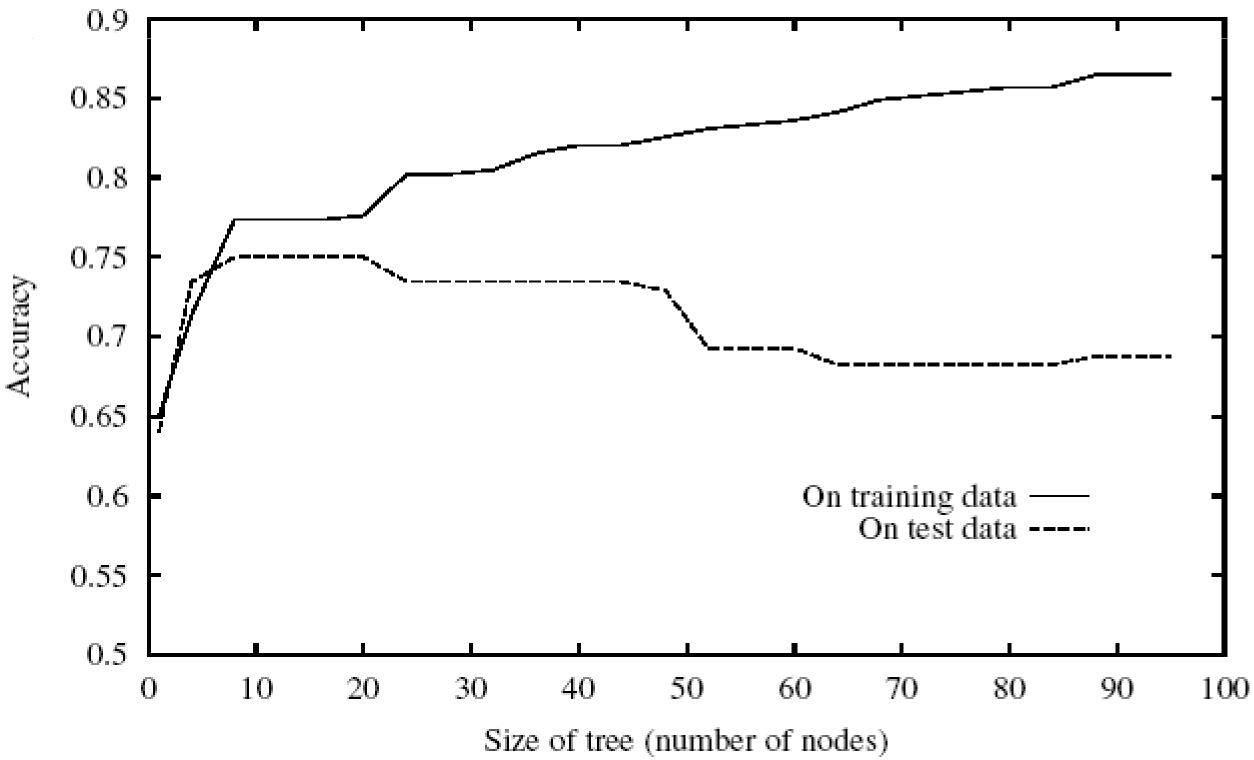
William of Ockham, born in the village of Ockham in Surrey (England) about 1285, was the most influential philosopher of the 14th century and a controversial theologian.

Otázky související s ID3

- Jak velké stromy konstruovat? Až do pokrytí všech příkladů? Co s přeučením?
- Spojitý definiční obor atributů
- Metody volby nejvhodnějšího atributu
- Atributy o různých cenách
- Chybějící hodnoty
- … ???

Přeučení

- Nechť **H** je prostor hypotéz. Hypotéza **h** ∈ **H** je přeučená, pokud exituje jiná hypotéza **h1** ∈ **H** taková, že chyba **h** na trénovacích datech je menší než chyba **h1**, avšak na celém prostoru instancí uvažovaných objektů je chyba **h1** menší než chyba **h**

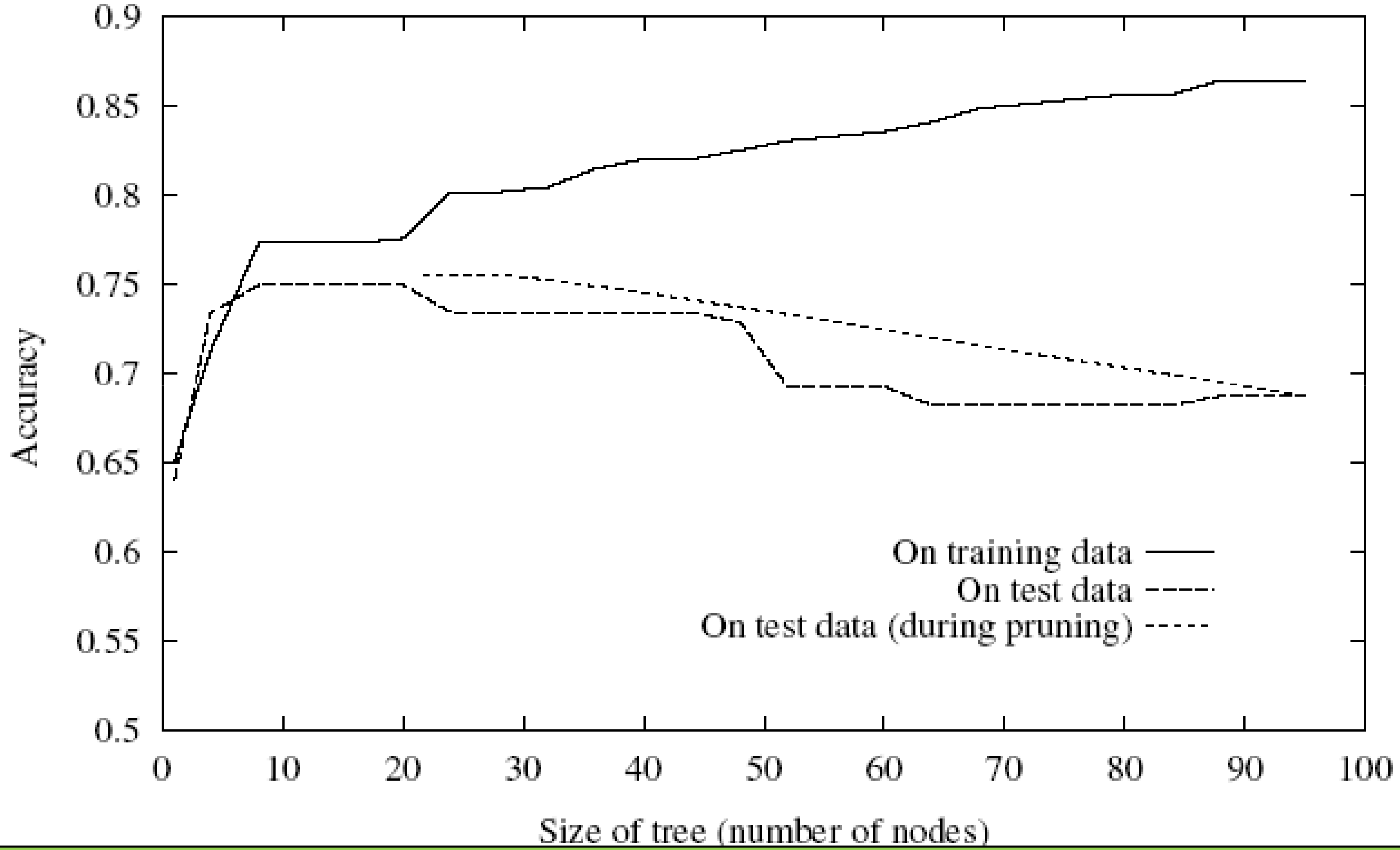


Často pozorovaná vlastnost zkonstruovaných stromů

Jak se vyhnout přeučení?

- Jak zvolit správnou velikost stromu?
- **Jak takový strom získat?**
  1. Zastavit růst stromu dřív než jsou vyčerpána všechna trénovací data
  2. **Prořezávání hotového stromu** – ukazuje se jako zvlášť užitečné! Volba vhodného prořezání pomocí **validační množiny dat** (vybraná nezávisle, tedy bez náhodných vlivů případně přítomných v trénovacích datech).
- **Algoritmus prořezávání „redukce chyby":**
  - Vyberte uzel, odstraňte podstrom, v něm začínající a přiřaďte většinovou klasifikaci.
  - Pokud se chyba na validačních datech zmenšila proveďte uvedené proříznutí (ze všech možností vyberte tu s největším zlepšením).

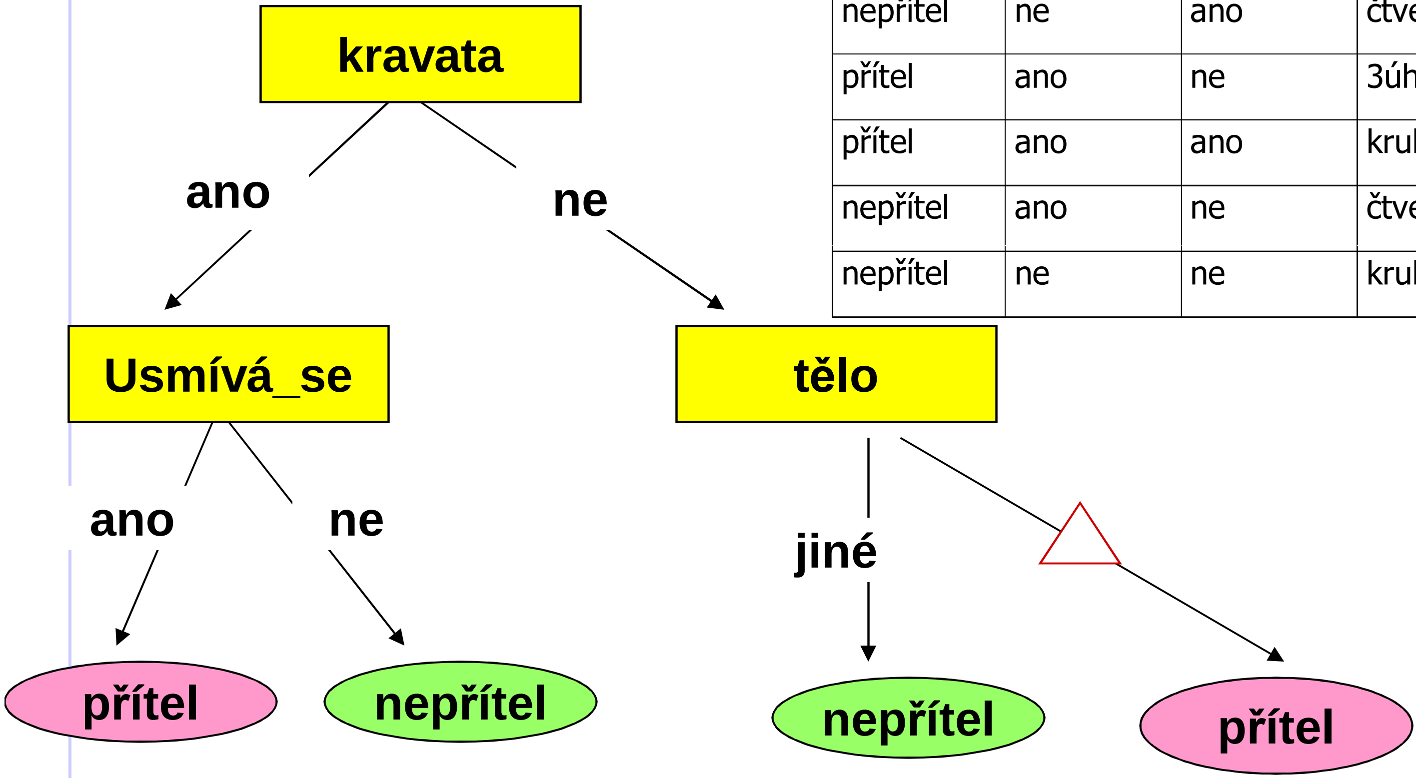
Hodnocení přesnosti klasifikace v závislosti na složitosti použitého stromu



Výsledky na „hladké linii“ odpovídají stromům získaným prořezáním tak, jak bylo otestováno na validačních datech (jiná než testovací!!!)

Rozhodovací strom jako logický výraz

Klasifikace	Usmíva_se	kravata	tělo	hlava	v_ruce
přítel	ano	ano	kruh	3úhelník	nic
přítel	ne	ne	3úhelník	3úhelník	balon
nepřítel	ne	ano	čtverec	3úhelník	balón
nepřítel	ne	ano	čtverec	kruh	meč
přítel	ano	ne	3úhelník	kruh	květ
přítel	ano	ano	kruh	kruh	nic
nepřítel	ano	ne	čtverec	čtverec	nic
nepřítel	ne	ne	kruh	kruh	meč



(Kravata=ano & usmívá_se=ano) V (Kravata=ne & tělo=3úh.) -> přítel

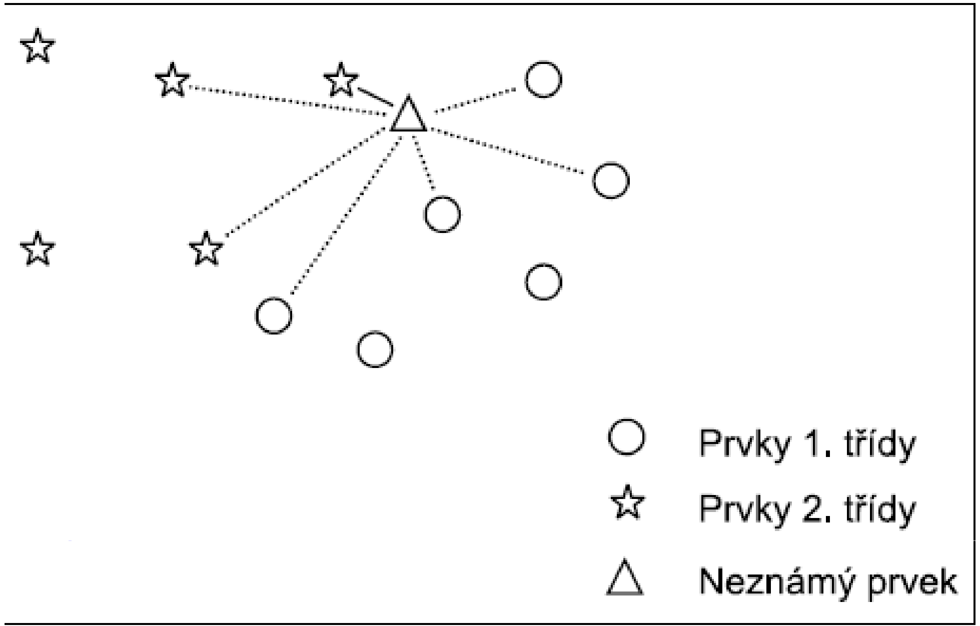
Závěrečné prořezávání pravidel (rule post-pruning) použité v **C4.5**

1. Vytvořte přeučený rozhodovací strom
2. Zapište výsledný strom ve tvaru disjunkce pravidel (každá větev = jedno pravidlo)
3. Každé jednotlivé pravidlo co nejvíc prořežte (odstraní se postupně ty podmínky, které nezhorší jeho klasifikační přesnost)
4. Uspořádejte výsledná pravidla podle jejich odhadnuté přesnosti a dále je používejte jako rozhodovací seznam

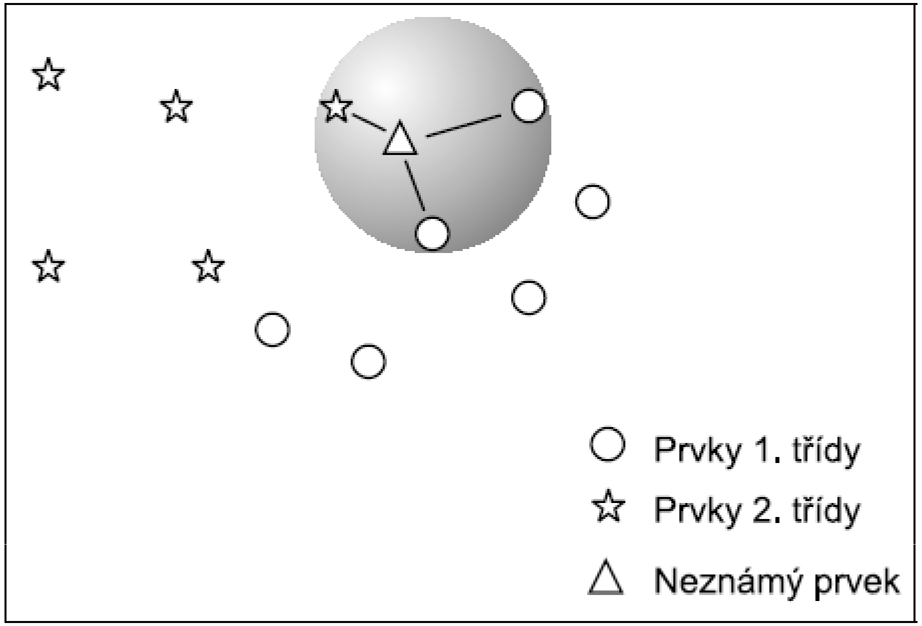
Odhad přesnosti pravidla

- na validační množině (= relativní počet správných závěrů)
- na trénovacích datech (= „pesimistický odhad počtu správých závěrů za předpokladu binomického rozdělení“)

Metoda *n* nejbližších sousedů



Obrázek 1: Popis klasifikace 1-NN



Obrázek 2: Popis klasifikace 3-NN

- Pro nový objekt je vypočtena vzdálenost od všech objektu v trén. příkl.
- Je nalezeno všech ***n*** trén. příkladů (= množina T^n), které jsou k novému objektu nejblíž. Nový objekt získá klasifikaci, která je v T^n nejčastější.
- Možné zobecnění: hledá se nejlepší vážicí koeficient pro jednotlivé atributy.

Příklad: Létání na simulátoru F16

Úkol: sestavit řídící systém pro ovládání leteckého simulátoru F16 tak, aby splnil předem definovaný plán letu daný takto:

1. vzlet a výstup do výšky 2000 stop
2. let v dané výšce směrem N do vzdálenosti 32000 stop od místa startu
3. zahnout vpravo v kurzu 330°
4. ve vzdálenosti 42000 stop od místa startu (ve směru S-N) provést obrat vlevo a zamířit zpět do místa startu (obrat je ukončen při kurzu mezi 140° a 180°)
5. vyrovnat směr letu s přistávací dráhou, tolerance 5° pro kurz a 10° pro výchylku křídel oproti horizontu
6. klesat směrem k počátku přistávací dráhy
7. přistát

Trénovací data: 3x30 letů (od 3 pilotů). Každý let popsán pomocí 1000 záznamů (poloha a stav letounu, pilotem provedený řídící zásah)

Záznam: Poloha a stav

on_gound boolean: je letadlo na zemi?

g_limit boolean: je překročen g limit letadla?

wing_stall boolean: je letadlo stabilní?

twist **integer**: 0°-360°, výchylka křídel vůči obzoru

elevation integer: 0°-360°, výchylka trupu vůči obzoru

azimuth integer: 0°-360°, směr letu

roll_speed integer: 0°-360°, rychlost změny výchylky křídel [°/s]

elev_speed integer: 0°-360°, rychlost změny výchylky trupu [°/s]

azimuth_speedinteger: 0°-360°, rychlost změny kurzu [°/s]

airspeed integer: rychlost letadla v uzlech

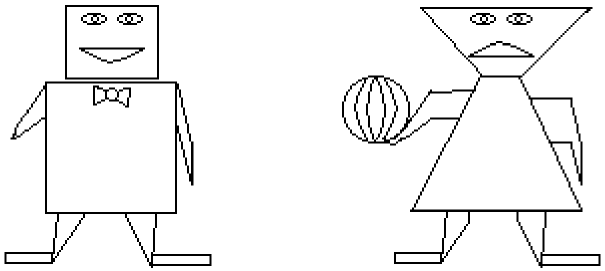
climbspeed integer: rychlost změny výšky [stop/s]

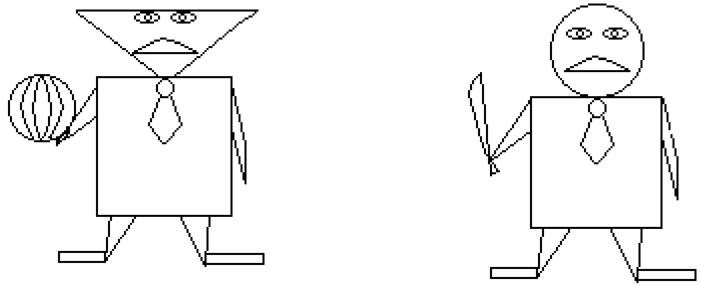
Záznam: Poloha a stav + **Řízení**

E/W distance	real: vzdálenost ve směru východ-západ od místa startu
N/S distance	real: vzdálenost ve směru sever-jih od místa startu
fuel	integer: váha paliva v librách
Řízení:	
rollers	real: nastavení ovladače horizontálního vychýlení
elevator	real: nastavení ovladače vertikálního vychýlení
thrust	integer: 0-100%, plyn
flaps	integer: 0°, 10° nebo 20°, nastavení křídlových lopatek

Každá ze 7 fází letu vyžaduje vlastní typ řízení (jiné zásahy pilota): trénovací příklady rozděleny do 7 odpovídajících skupin. V každé skupině je zkonstruován zvlášť rozhodovací strom pro každý typ řídícího zásahu (rollers, elevator, thrust, flaps), t.j. *7 x 4 stromů*

Zkuste navrhnout nejjednodušší klasifikační algoritmus





přátelští

nepřátelští