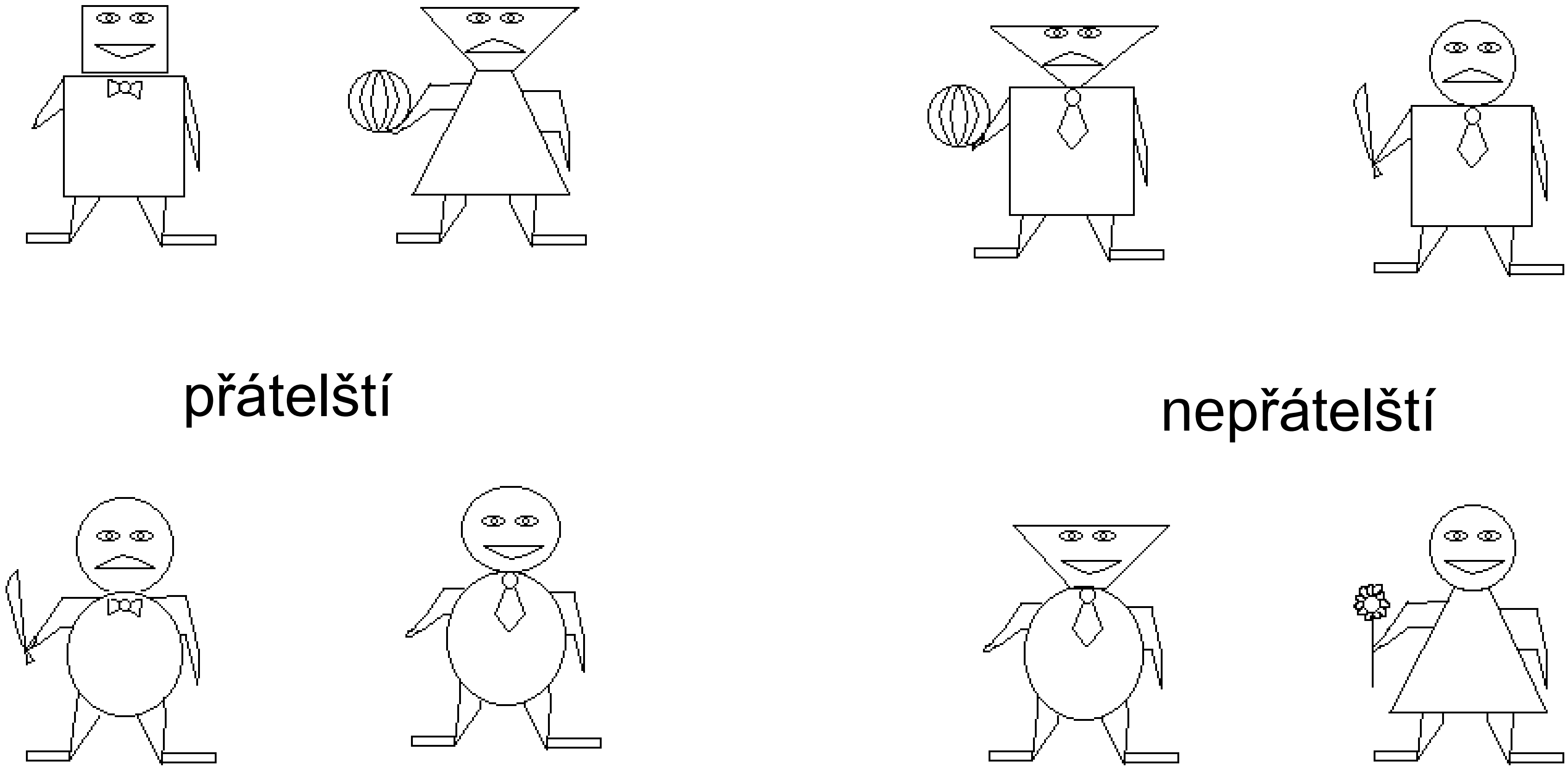
Rozhodovací stromy a jejich konstrukce z dat

Nature Inspired
Technologies Group

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

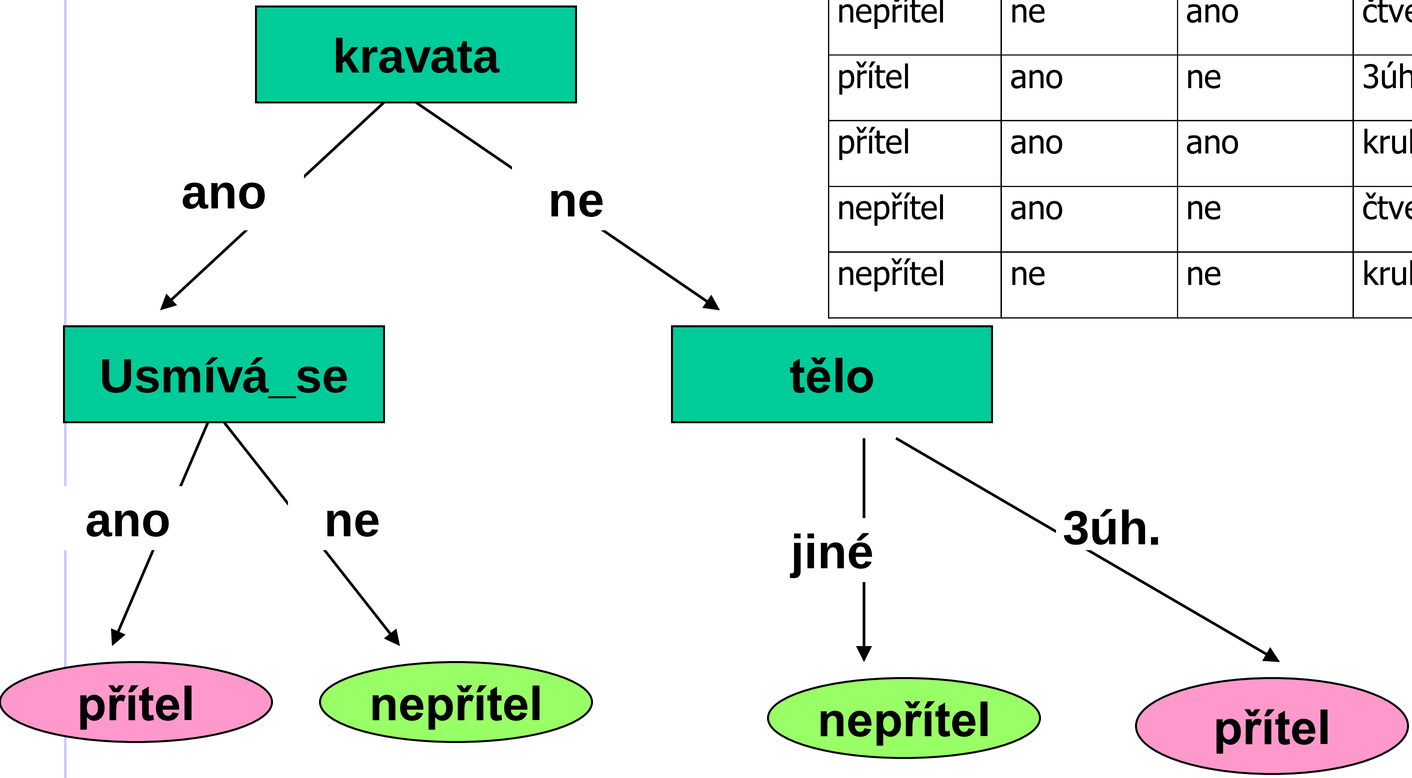


Příklad 1: Roboti a atributový popis

tvar hlavy	*úsměv*	*ozdoba krku*	*tvar těla*	*předmět v ruce*	*přátelský*
Kruh	ne	kravata	čtverec	šavle	ne
Čtverec	ano	motýlek	čtverec	nic	ano
Kruh	ne	motýlek	Kruh	šavle	ano
Trojúhelník	ne	kravata	čtverec	balón	ne
Kruh	ano	nic	trojúhelník	květina	ne
Trojúhelník	ne	nic	trojúhelník	balon	ano
Trojúhelník	ano	kravata	Kruh	nic	ne
Kruh	ano	kravata	Kruh	nic	ano

Rozhodovací strom 1
pro danou množinu příkladů

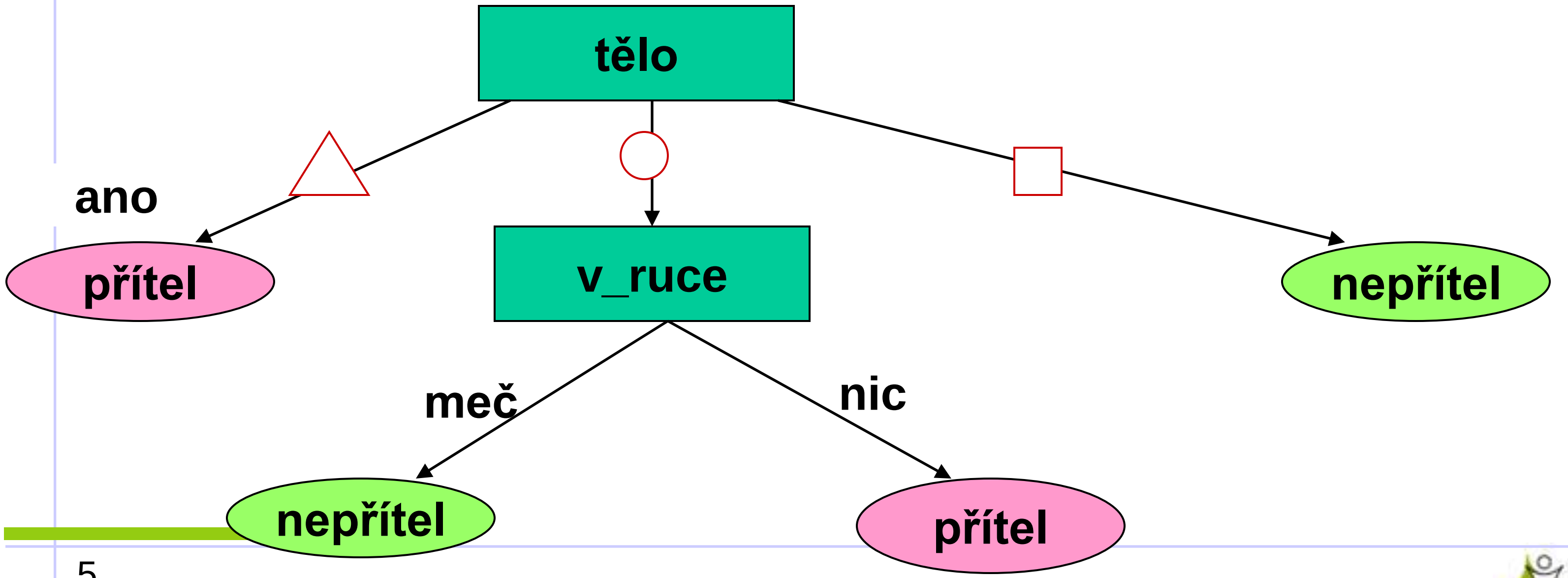
Klasifikace	Usmíva_se	kravata	tělo	hlava	v_ruce
přítel	ano	ano	kruh	3úhelník	nic
přítel	ne	ne	3úhelník	3úhelník	balon
nepřítel	ne	ano	čtverec	3úhelník	balón
nepřítel	ne	ano	čtverec	kruh	meč
přítel	ano	ne	3úhelník	kruh	květ
přítel	ano	ano	kruh	kruh	nic
nepřítel	ano	ne	čtverec	čtverec	nic
nepřítel	ne	ne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



Rozhodovací strom 2 pro tutéž množinu příkladů

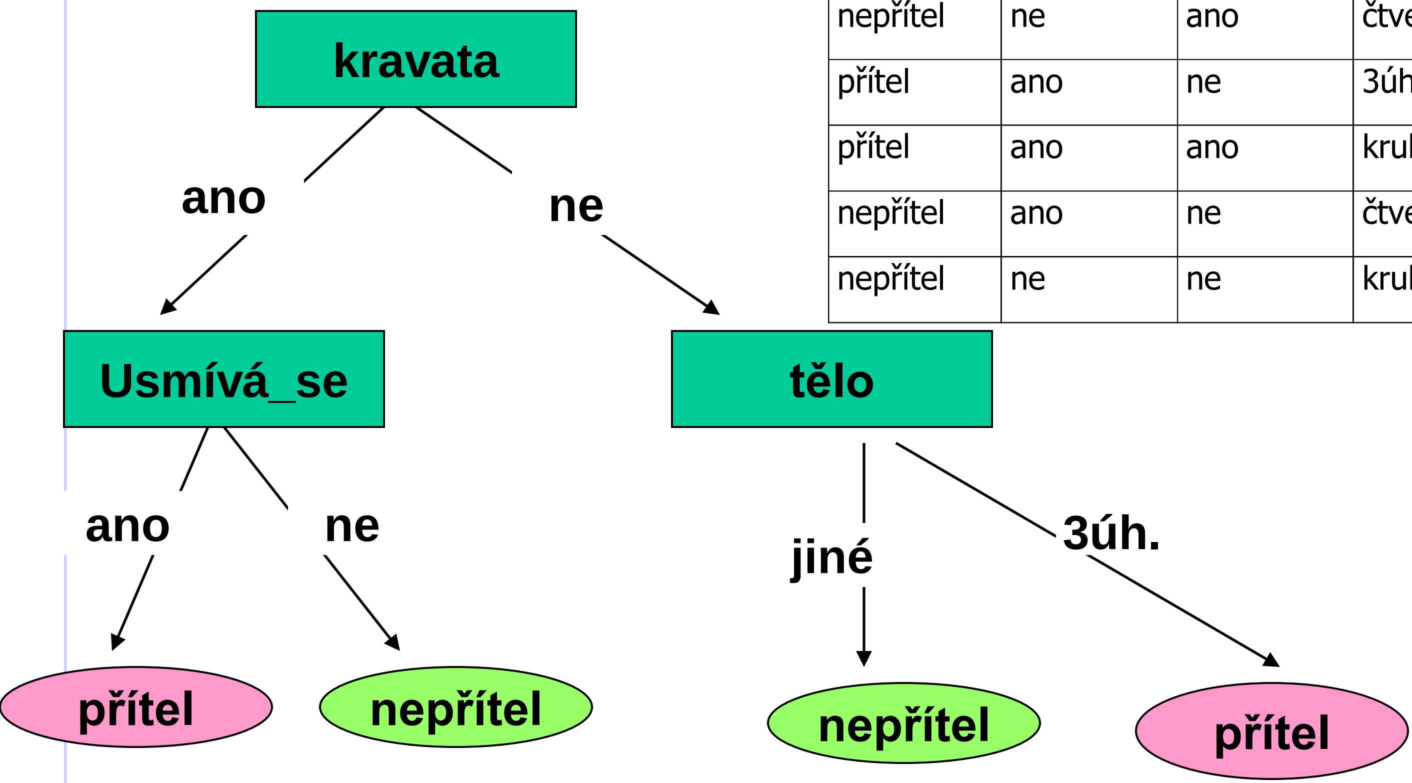
Klasifikace	Usmíva_se	kravata	tělo	hlava	v_ruce
přítel	ano	ano	kruh	3úhelník	nic
přítel	ne	ne	3úhelník	3úhelník	balon
nepřítel	ne	ano	čtverec	3úhelník	balón
nepřítel	ne	ano	čtverec	kruh	meč
přítel	ano	ne	3úhelník	kruh	květ
přítel	ano	ano	kruh	kruh	nic
nepřítel	ano	ne	čtverec	čtverec	nic
nepřítel	ne	ne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

Který strom je lepší?



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

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

	Usmívá_se	Kravata	tělo=3úhel.	hlava=3úh.	v_r.=nic
Klasifikace	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 **at_0** (t.j. atribut, jehož hodnoty nejlépe diskriminují mezi pozitivní a neg. příklady) a tím ohodnoť kořen vytvářeného stromu.

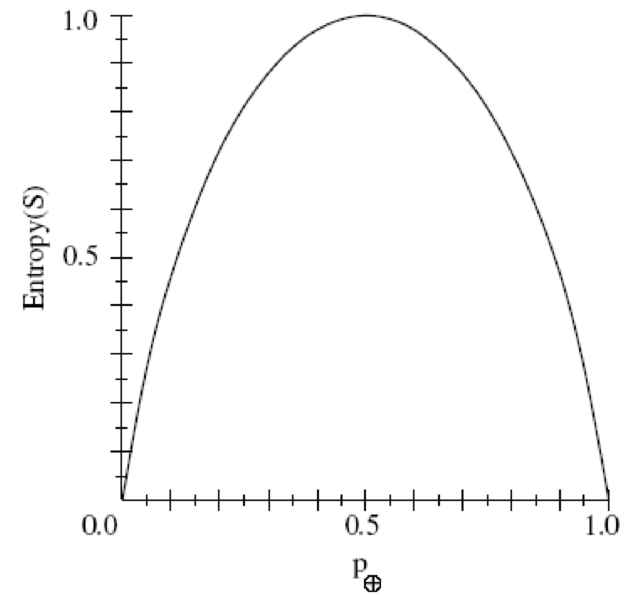
2. Rozděl množinu **S** na podmnožiny **$S_1, S_2, \ldots, S_n$** podle hodnot atributu **at_0** a pro každou množinu příkladů **S_i** vytvoř nový uzel jako následníka právě zpracovávaného uzlu (kořenu)

3. Pro každý nově vzniklý uzel s přiřazenou podmnožinou **S_i** proveď:

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

Základní algoritmus ID3

- Realizuje prohledávání prostoru všech možných (vzhledem k jazyku trénovacích dat)
  - shora dolů
  - s použitím hladové strategie
- Volba atributu pro větvení na zákl. charakterizace „(ne)homogenity vzniklého pokrytí" : **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 zisku (*gain*)

Co se stane, pokud některý atribut má „skoro“ unikátní hodnotu pro každý trénovací příklad?

$\mathrm{Gain}(S,at) = E(S) - \mathbf{E(S,at)} = \mathbf{E(S)} - \sum_{i=1}^{n} |S_i|/|S| * \mathbf{E(S_i)}$, kde

$\mathbf{E(S_i)}$ je skoro 0

Tento agrument je vybrán jako nejlepší.

Je opravdu užitečný pro testovací data?

Neměla by se taková situace nějak „penalizovat“?

JAK?

Jak charakterizovat rozklad množiny *S* na disjunktní podmnožiny S_i podle 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. při $|S_i| = 1$, je roven ($\log_2 n$)

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

Volba atributů a další „speciální situace“

❖ Reálné hodnoty.

→ používá se **diskretizace**

❖ Různé ceny pro získání hodnoty atributu

❖ Atribut má příliš rozsáhlý definiční obor

→ používá se **změna kriteria pro volbu atributů**

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 InfoGain, 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)

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

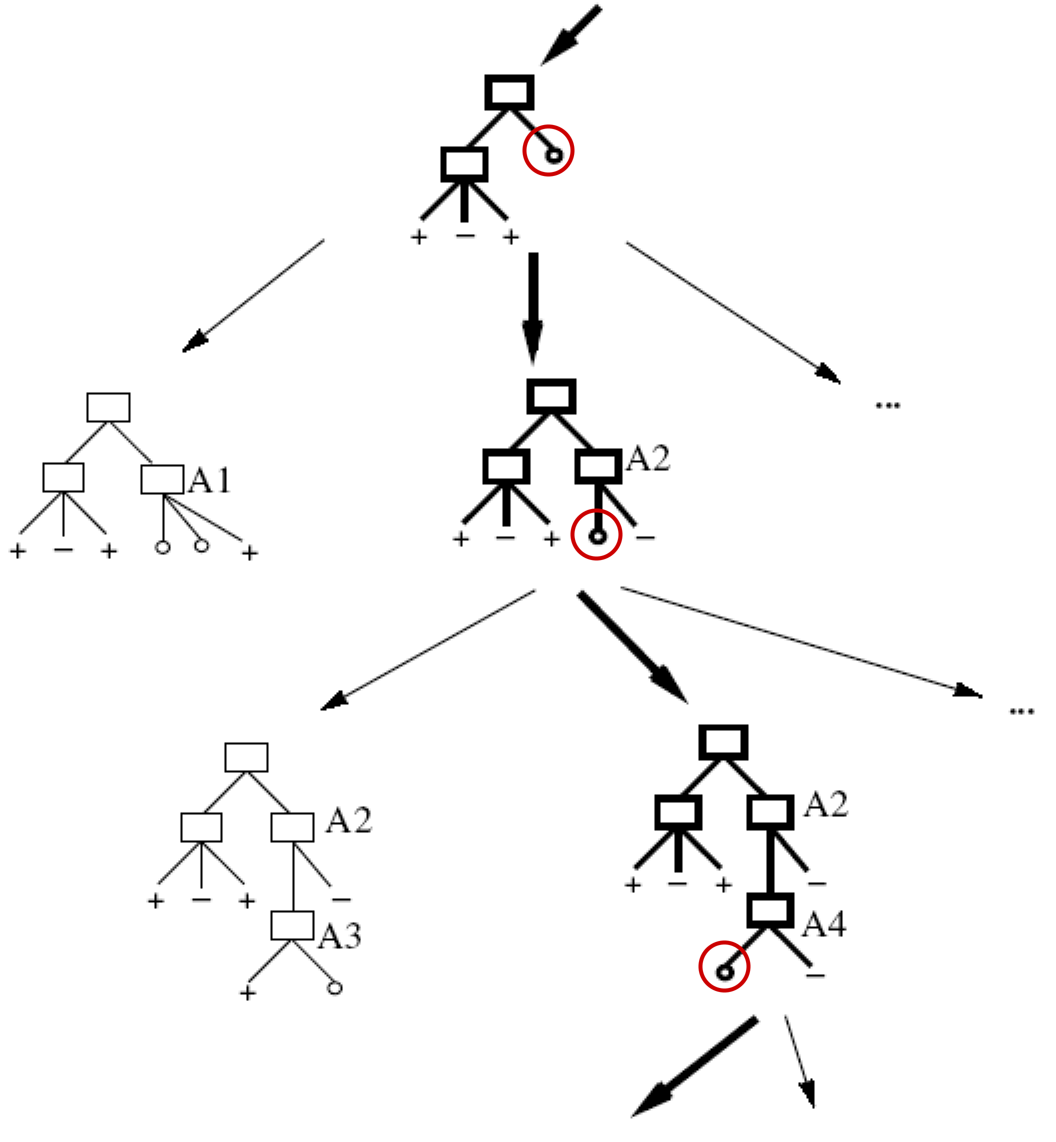
  - Nunez (1988)

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

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)**

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), *na rozdíl např. od prostoru verzí* --> existuje mnoho hypotéz konzistentních s daty!

- Aktuální **množina hypotéz je vždy jednoprvková** (hladová volba následníka), nelze jej tedy použít pro odpověď na dotaz „kolik existuje 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

Bias = popis podmínek vedoucích k výběru výsledné hypotézy

- **Bias pro ID3** je důsledkem použité metody prohledávání a **je typicky preferenční**: Algoritmus preferuje
  - stromy o nižší hloubce a
  - Stromy, které umísťují atributy s vyšším Informačním ziskem blíž ke kořeni
- Naopak **bias algoritmu prostoru verzí** používá omezený prostor hypotéz -- > jedná se o **jazykový bias** (restrikční b.), jehož nebezpečí je, že nedokáže reprezentovat cílovou funkci

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!

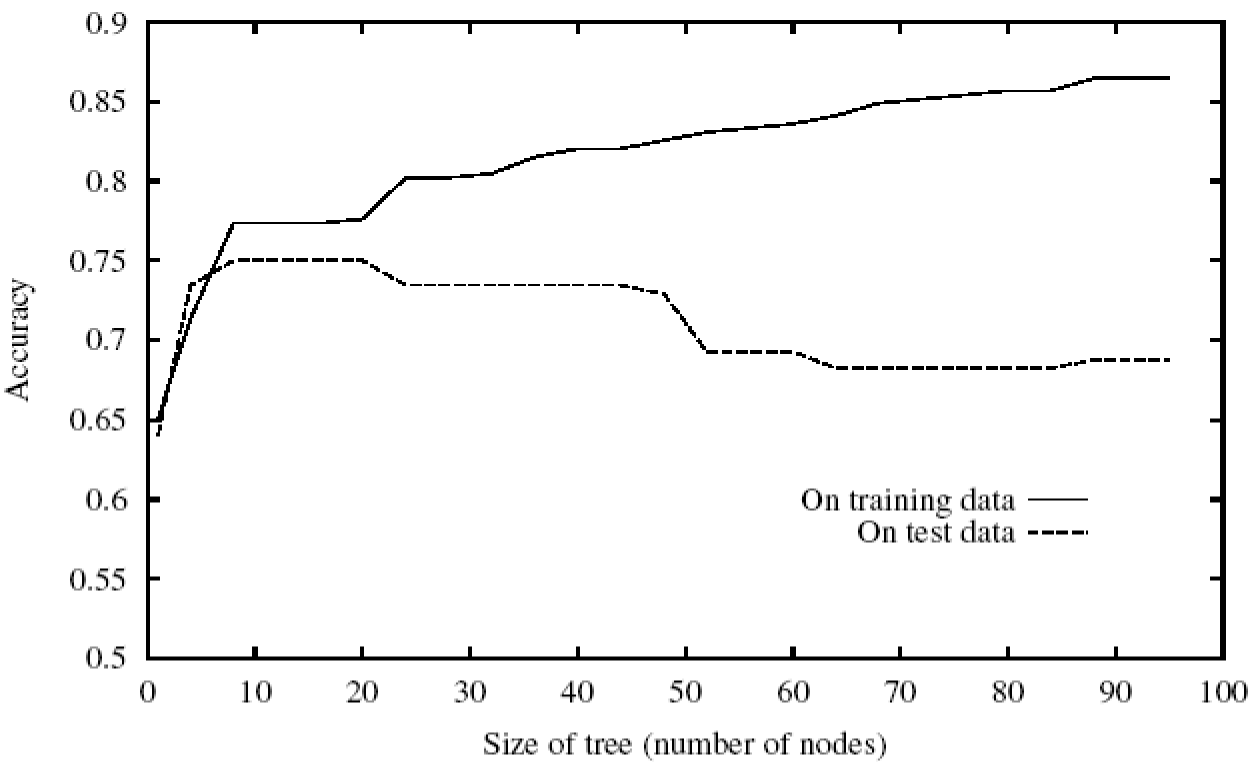
William of Ockham, born in the village of Ockham in Surrey (England) about 1285, was the most influential philosopher of the 14th century and a controversial theologian.

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

- Necht’ **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**

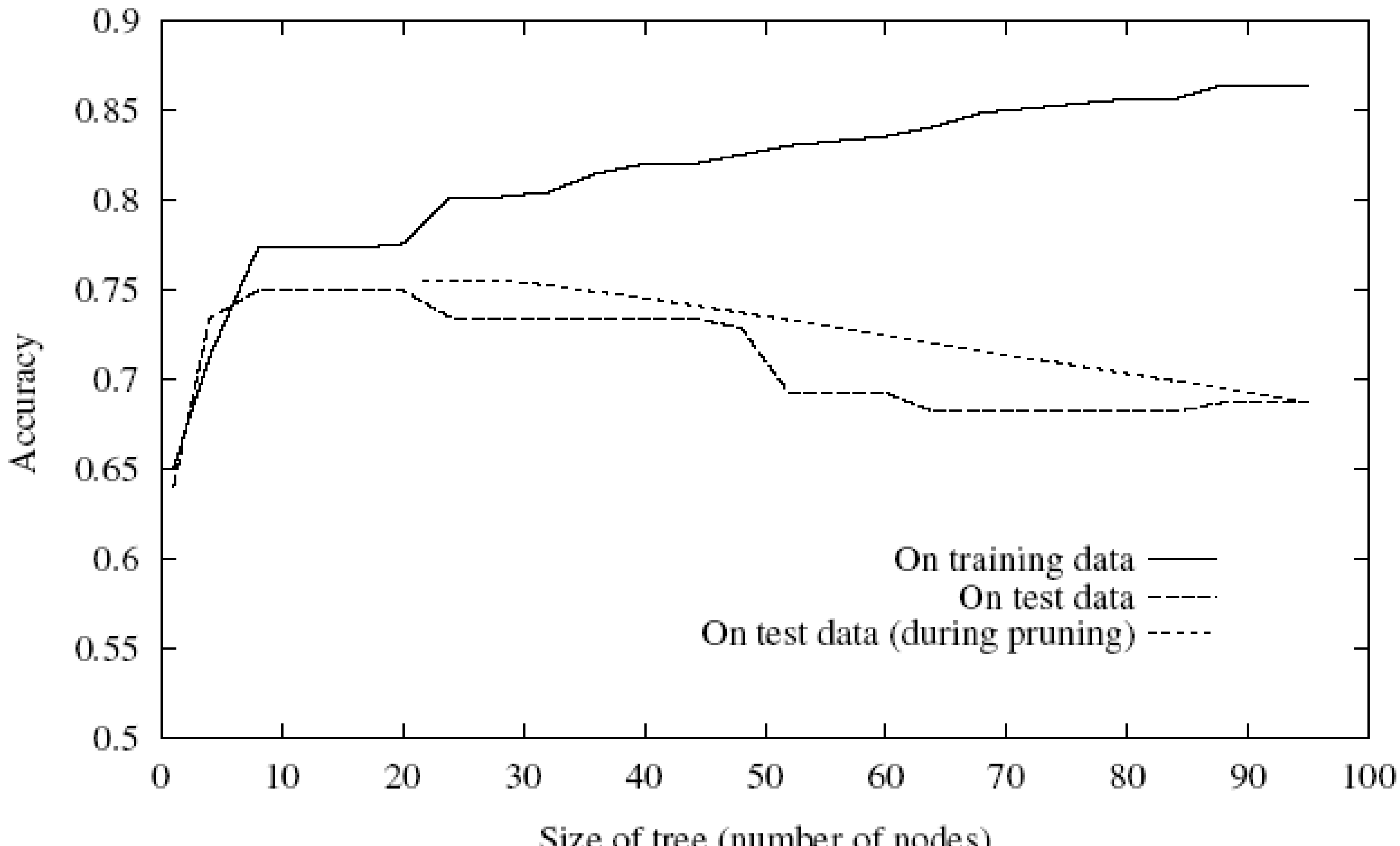


Často pozorovaná vlastnost zkonstruovaných stromů

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!!!)

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 klasifikační přesnost)
4. Uspořádejte výsledná pravidla podle jejich odhadnuté přesnosti a dále je používejte jako rozhodovací seznam

Odhad přesnosti pravidla

- Validační množina
- Na trénovacích datech jako „pesimistický odhad na trénovacích datech za předpokladu binomického rozdělení“

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*