Rozhodovací stromy a jejich konstrukce z dat

Co je rozhodovací strom?

Rozhodovací (klasifikační) strom:

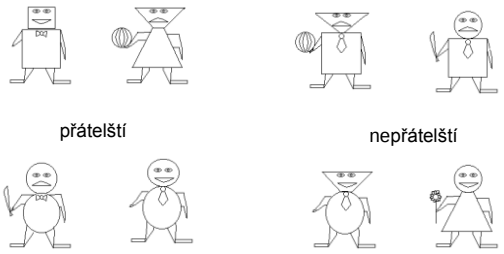
- funkce, které dáme na vstup vektor atributů a která vrátí „rozhodnutí" – jednu z možných výstupních hodnot.
- k rozhodnutí dochází sekvencí testů
- každý vnitřní uzel stromu reprezentuje podmínku, každý list stromu reprezentuje rozhodnutí

Příklady rozhodovacích stromů:

- určovací klíče v biologii
- diagnostická sekce v „Domácím lékaři"
- pomoc s řešením problémů ve Windows

Jak ale strom vytvořit, aby doba do rozhodnutí byla malá?

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

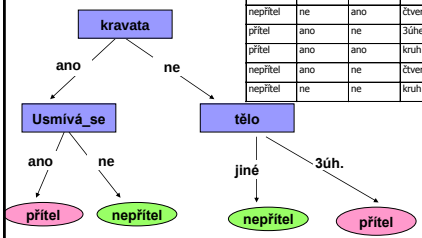


Příklad 1: Roboti a atributový popis

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

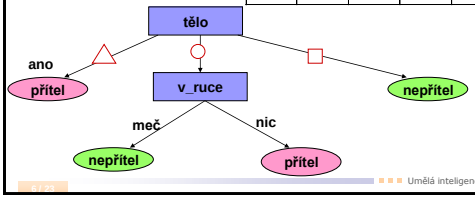
Klasifikace	Usmíva_se	kravata	tělo	hlava	v_ruce
přítel	ano	ano	kruh	3úhelník	nic
přítel	ne	ne	3úhelník	3úhelník	balon
nepřítel	ne	ano	čtverec	3úhelník	balón
nepřítel	ne	ano	čtverec	kruh	meč
přítel	ano	ne	3úhelník	kruh	květ
přítel	ano	ano	kruh	kruh	nic
nepřítel	ano	ne	čtverec	čtverec	nic
nepřítel	ne	ne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



Rozhodovací strom 2 pro tutéž množinu příkladů

Který strom je lepší?

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



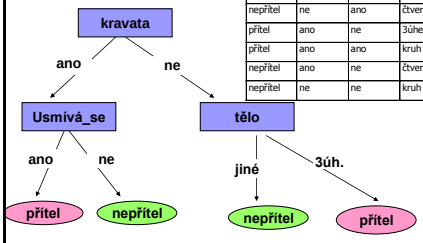
Který strom je lepší?

Těžko říct. Co znamená „lepší"?

- **Přesnější**? Oba jsou stejně přesné.
- **Jednodušší**? Oba stromy popisují datovou sadu stejně přesně, mají hloubku 2 a 4 listy. Strom 1 obsahuje 3 podmínky, strom 2 jen 2.
- Co když je jeden model jednodušší, ale méně přesný?
- Co když jsou v datech chyby nebo obsahují šum?
- **Úspěšnější při aplikaci na nová data**? Žádná zatím nemáme. Potřebovali bychom sadu testovacích dat (jiná sada dat z téhož zdroje).

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

Expresivita rozhodovacích stromů

- Každý binární rozhodovací strom je ekvivalentní nějakému výrazu v DNF.
- Jakákoli funkce ve výrokové logice se dá vyjádřit jako rozhodovací strom.
- Pro některé z funkcí jsou stromy vhodné, pro jiné ne.
- Jak velká je množina booleovských funkcí nad *n* atributy?
  - Počet pravdivostních tabulek, které lze s *n* atributy vytvořit.
  - Tabulka má 2^n řádků → 2^(2^n) funkcí.
  - Stromů je ještě víc, jedna funkce může být vyjádřena více stromy.
- Potřebujeme nějaký chytrý algoritmus, který by z tolika hypotéz (stromů) vybral nějakou dobrou. Jak na to???

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

Jak zvolit "nejlepší" atribut?

Rozdělme množinu **S** na podmnožiny $S_1, S_2, ..., S_n$ na základě hodnot diskrétního atributu **at.**

Měření množství informace uvnitř S_i def. pomocí entropie (Shanon)

$$E(S_i) = -(p_i^+)*\log p_i^+ - (p_i^-)*\log p_i^-,$$

kde (p_i^+) je pravděpodobnost, že libovolný příklad v S_i je pozitivní; hodnota (p_i^+) se odhaduje jako odpovídající frekvence (četnost).

Celková entropie **E(S,at)** tohoto systému je

$$E(S,at) = \sum_{i=1}^{n} P(S_i) * E(S_i),$$

kde $P(S_i)$ je pravděpodobnost události S_i, tj. poměr $|S_i| / |S|$.

Vypočteme **E(S,at)** pro všechny **at** a zvolíme atribut at_0 s minimální entropií $E(S,at_0)$.

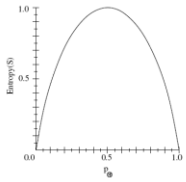
Indukce stromu z trénovacích dat

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 $S_1, S_2, ..., S_n$ 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 v S_i nejsou žádné případy, pak je uzel listem a pro rozhodnutí použij rodiče. Konec.
   - Jestliže všechny příklady v S_i patří do stejné třídy, pak je uzel listem a jeho rozhodnutí bude třída, do níž příklady patří. Konec.
   - Jestliže nezbyl atribut, podle něhož se můžeme rozhodovat, a S_i přesto obsahuje příklady více tříd (chyby v datech, šum), pak je uzel listem a jako rozhodnutí použij majoritní třídu z S_i.
   - jinak pokračuj ve větvení: jdi na bod **1** s tím, že S : = S_i.

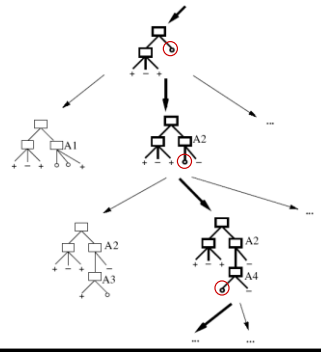
Základní algoritmus ID3

- Realizuje prohledávání prostoru všech možných (vzhledem k jazyku trénovacích dat) stromů:
  - 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



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

Postup prohledávání

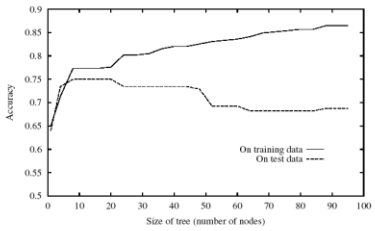


Vlastnosti ID3 – důsledky postupu prohledávání

- 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 (backtracking) --> **možnost uváznutí v lokálním optimu**
- **Hypotéza se vytváří na základě všech příkladů** (nikoliv inkrementálně) --> metoda není příliš ovlivněna šumem

Přeučení

- Nechť **H** je prostor hypotéz. Mějme hypotézy **h** a $\mathbf{h_1} \in \mathbf{H}$, označme jejich chybu na trénovacích datech jako $\mathbf{e_{train}}$ a $\mathbf{e_{train1}}$ a chybu na testovacích datech jako $\mathbf{e_{test}}$ a $\mathbf{e_{test1}}$.
- Hyp. **h** je přeučená, pokud pro nějakou $\mathbf{h_1}$: $\mathbf{e_{train}} < \mathbf{e_{train1}}$, avšak $\mathbf{e_{test}} > \mathbf{e_{test1}}$.

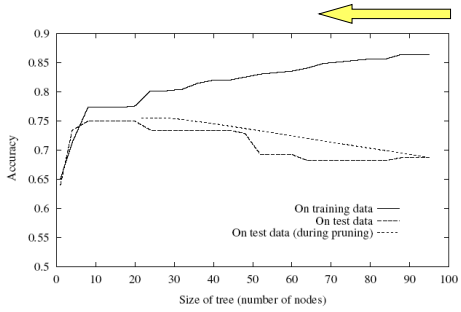


Přeučení se často projevuje u všech druhů modelů, nejen u stromů.

Jak se vyhnout přeučení?

1. **Brzké zastavení:** nestavět strom o plné hloubce (např. chí-kvadrát test: „Stojí toto větvení opravdu za to?“). Špatně řeší situace, kdy žádný z atributů není vhodný pro větvení, ale oba jsou důležité. Prořezání si poradí lépe.
2. **Prořezávání hotového stromu:** Volba vhodného prořezání pomocí **validační množiny dat** (vybraná nezávisle, tedy bez náhodných vlivů případně přítomných v trénovacích datech) nebo pomocí **chí-kvadrát testu**.

- **Algoritmus prořezávání „redukce chyby“:**
  - Vyberte uzel, odstraňte podstrom, v něm začínající a přiřaďte většinovou klasifikaci.
  - Pokud se chyba na validačních datech zmenšila proveďte uvedené prořízntutí (ze všech možností vyberte tu s největším zlepšením).



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

Chybějící hodnoty

- Stromy jsou jedním z mála typů modelů, které se dokáží vypořádat s chybějícími hodnotami.
- Jak klasifikovat objekt, jemuž chybí hodnota atributu, kterou bychom potřebovali pro provedení testu v listu *L*?
  - Představíme, že objekt má všechny možné hodnoty daného atributu.
  - Sledujeme všechny možné cesty až do listů.
  - Rozhodnutí listů vážíme podle četnosti hodnot atributu v listu *L*.
- Počítáme s chybějícími hodnotami už při učení a zavedeme je jako další hodnotu (např. „Ano", „Ne", „Neuvedeno")

Atributy s mnoha možnými hodnotami

Co když mezi atributy bude např. jméno?

- Jiná hodnota pro každý případ, informační zisk takového atributu je roven entropii celé datové sady!
- Atribut je vybrán do kořene, ale takový strom je k ničemu!

Řešení: **Gain ratio**

- Normalizace inf. zisku maximálním množstvím informace, kterou atribut **A** může přinést.

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

- *SplitInformation* odpovídá entropii rozdělení **S** podle všech hodnot atributu **A**.

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

Atributy s různou cenou

- 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)

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

Spojité vstupní příznaky

- Nekončná množina hodnot (nekonečně mnoho větví?)
- Používá se binární větvení: x < K, x >= K, K je bod větvení (split point)
- Volí se K s největším informačním ziskem
- Uspořádejte příklady podle velikosti zpracovávaného atributu a jako kandidátní K zvolte takové hodnoty, které leží mezi hodnotami, 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

Spojitá výstupní proměnná

Regresní strom

- V každém listu konstanta nebo lineární funkce podmnožiny numerických atributů
- Algoritmus učení musí rozhodnout, kdy přestat větvit a vytvořit list s lineární funkcí