Dobývání znalostí

Doc. RNDr. Iveta Mrázová, CSc.

Katedra teoretické informatiky
Matematicko-fyzikální fakulta
Univerzity Karlovy v Praze

Dobývání znalostí

– Rozhodovací stromy –

Doc. RNDr. Iveta Mrázová, CSc.

Katedra teoretické informatiky
Matematicko-fyzikální fakulta
Univerzity Karlovy v Praze

Rozhodovací stromy
(decision trees - motivace)

Chudý na telefonu? V britské bance ho přepojí do Indie

Londýn - Až si klient britského bankovního domu Barclays v dubnu zavolá do své banky, prolustruje ho ještě během vyzvánění „elektronický vrátný". Bohatší klienty přepojí k poradcům v Británii, na méně majetné čeká pracovník call-centra v Indii.

Podle britského deníku Daily Mail si banka od nového telefonního síta slibuje odfiltrování nemajetných zákazníků, kteří nemají na to, aby si kupovali určité bankovní produkty, od těch, kterým je naopak možné služby prodat. „Předtím nám volali především lidé, kteří měli přečerpaný úvěr, těm nebylo možné prodat vůbec nic," řekl deníku Daily Mail jeden z obchodníků banky Barclays, který si přál zůstat v anonymitě.

Kritici však banku za takovýto postup pranýřují – považují ho za diskriminaci milionů klientů, kteří chtějí jen zjistit informace o svém účtu. „Je to segregační systém, který zřetelně preferuje bohatší zákazníky banky," tvrdí Eddy Weatherill z Nezávislé bankovní dozorčí služby (IBAS). *Pokračování na str. B2*

Rozhodovací stromy
(decision trees - motivace)

Chudý na telefonu? V bance ho přepojí do Indie

Pokračování ze str. B1

Podle něj tak nejde o to, pomoci lidem s jejich dotazy, nýbrž vydělávat co nejvíce peněz.

„Zákazníci nově budou hovořit s různými týmy na různých místech, a to v závislosti na povaze jejich dotazu," nechal se slyšet v britském tisku mluvčí banky Barclays.

Systém ověří číslo volajícího a automaticky o něm vyhledá dostupné informace – například výši zůstatku, zda dostal od banky úvěr, a pokud ano, tak jak velký. A pak na základě bankou stanovených kritérií rozhodne, kdo se volajícímu bude věnovat.

České banky většinou o podobné selekci zatím neuvažují. „Volací centrum máme jen jedno, pokud klient zavolá, je vždy obsloužen na této lince," říká například Tomáš Kofroň, mluvčí Raiffeisenbank. „K tomu máme speciální linky pro firemní klienty nebo pro blokaci karet. Pro obsluhu více bonitních klientů využíváme spíše osobních bankéřů," dodává Kofroň.

„V současné době neuplatňujeme žádný systém rozdílného či přednostního obsloužení podle bonity klientů," potvrzuje Kristýna Havligerová, mluvčí České spořitelny.

Podobný princip selekce volajících zákazníků tradičně uplatňují někteří mobilní operátoři. I jejich telefonický systém hned po zavolání většinou zjistí, jakou měsíční útratu a také platební morálku daný klient má. Podle toho pak klienta přepojí.

Jiný zákazník, jiný poradce

„Naším cílem je vyřešit požadavky všech zákazníků formou segmentované péče," vysvětluje Martina Kemrová, mluvčí společnosti T-Mobile. Podle ní má každá kategorie zákazníků jiné požadavky a také nabízené služby se pro ně mohou lišit. Například zákazníci s předplacenými kartami se nejdříve dovolají na hlasový automat.

„Avšak prémioví zákazníci jsou pak spojeni v nejkratší možné době rovnou s operátorem," vysvětluje Kemrová. Dodává, že péči pro nižší segmenty, tedy zákazníky s nižší útratou, pro ně částečně dělají externí firmy.

„Nabídka služeb pro tyto segmenty zákazníků je jednodušší. U vyšších segmentů, tedy tam, kde jde o složitější služby a produkty, zajišťujeme péči sami," uzavírá mluvčí Martina Kemrová.

PAVEL P. NOVOTNÝ

Rozhodovací stromy (decision trees)

- Řešení klasifikačních úloh
- Vytvářený strom modeluje proces klasifikace
  - Efektivní vytváření stromů
  - Dělení příznakového prostoru do pravoúhlých oblastí
- Klasifikace vzorů podle oblasti, ve které se nacházejí

Rozhodovací stromy (2)

Definice:

Mějme databázi $D = \{\vec{t}_1, \ldots, \vec{t}_n\}$, kde $\vec{t}_i = (t_{i1}, \ldots, t_{in})$.
Dále mějme atributy $\{A_1, A_2, \ldots, A_n\}$ a množinu tříd $C = \{C_1, \ldots, C_m\}$.

Rozhodovací strom pro D je strom, kde:

- Každý vnitřní uzel je ohodnocen atributem A_i
- Každá hrana je ohodnocena predikátem (použitelným na atribut rodičovského uzlu)
- Každý list je ohodnocen třídou C_j

Rozhodovací stromy (3)

Řešení klasifikačních úloh pomocí rozhodovacích stromů vyžaduje dva kroky:

- Indukce rozhodovacího stromu
  - Podle trénovacích dat
- $\forall \vec{t}_i \in D$ použij rozhodovací strom a urči třídu

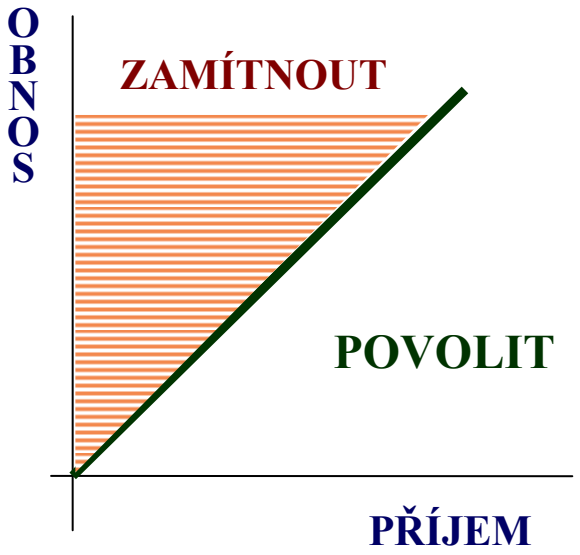
Výhody:

- Jednoduché
- Efektivní
- Extrakce jednoduchých pravidel
- Použitelné i pro velké databáze

Rozhodovací stromy (4)

Nevýhody:

- Obtížnější zpracování spojitých dat (kategorizace atributů ~ rozdělení příznakového prostoru do pravo-úhlých oblastí)
  → nelze použít vždy
  - Příklad: **Půjčka**
- Obtížné zpracování při chybějících údajích
- Přeučení („over-fitting“)
  → PROŘEZÁVÁNÍ
- Vzájemná korelace mezi atributy se nebere v úvahu



Rozhodovací stromy (5)

Atributy pro dělení:

- Ohodnocení uzlů vytvářeného stromu

Dělicí predikáty:

- Ohodnocení hran vytvářeného stromu

Ukončovací kritérium:

- Např. příslušnost všech vzorů z redukované množiny ke stejné třídě

Rozhodovací stromy (6)

Indukce rozhodovacího stromu – algoritmus:

VSTUP: D // trénovací data
VÝSTUP: T // rozhodovací strom

Vytvoření rozhodovacího stromu:
// základní algoritmus

T = {};
urči nejlepší kritérium pro dělení;
T = vytvoř kořen a ohodnoť ho atributem pro dělení;
T = přidej hranu pro každý dělicí predikát a ohodnoť ji;

Rozhodovací stromy (7)

Indukce rozhodovacího stromu – algoritmus:

// pokračování
FOR každou hranu DO
  D´ = databáze vytvořená použitím dělicího predikátu na D;
  IF ukončovací kritérium splněno pro danou cestu
    THEN
      T´ = vytvoř list a ohodnoť ho příslušnou třídou;
    ELSE
      T´ = vytvoření rozhodovacího stromu (D´);
 T = připoj T´ k hraně

Rozhodovací stromy (8)

Algoritmus TDIDT:

(~ Top Down Induction of Decision Trees)

- Indukce stromů metodou shora dolů (rozděl a panuj)
- **Algoritmus TDIDT:**
  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, opakuj pro tento uzel postup od bodu 1;
     jinak skonči

Rozhodovací stromy (9)

Výběr vhodného atributu pro větvení stromu:

- Cíl: vybrat atribut, který od sebe nejlépe odliší příklady z různých tříd
- Entropie ~ míra neuspořádanosti systému

$$H = -\sum_{t=1}^{T} \left(p_t \log_2 p_t \right)$$

p_t … pravděpodobnost výskytu třídy t (~ relativní četnost třídy t počítaná na určité množině příkladů)

t … počet tříd

(Pozn.: $\log_b x = \log_a x / \log_a b$)

Rozhodovací stromy (10)

Výpočet entropie pro jeden atribut:

- Pro každou hodnotu ***v*** , které může nabýt uvažovaný atribut ***A*** , spočítej entropii ***H (A (v))*** na skupině příkladů, které jsou pokryty kategorií ***A (v)***

$$H(A(v)) = -\sum_{t=1}^{T} \frac{n_t(A(v))}{n(A(v))} \log_2 \frac{n_t(A(v))}{n(A(v))}$$

- Spočítej **střední entropii** ***H (A)*** jako vážený součet entropií ***H (A (v))***

$$H(A) = \sum_{v \in Val(A)} \frac{n(A(v))}{n} H(A(v))$$

Rozhodovací stromy (11)

Použití rozhodovacího stromu pro klasifikaci nových případů:

- V nelistových uzlech stromu jsou uvedeny atributy použité při větvení
- Hrany stromu odpovídají hodnotám těchto atributů
- V listech stromu je informace o přiřazení ke třídě
- Od kořene stromu se postupně zjišťují hodnoty příslušných atributů

Rozhodovací stromy (12)

Převod stromu na pravidla:

- Každé cestě stromem od kořene k listu odpovídá jedno pravidlo
- nelistové uzly (atributy) se (spolu s hodnotou pro příslušnou hranu) objeví v předpokladech pravidla
- Listový uzel (cíl) bude v závěru pravidla

Rozhodovací stromy
Příklad: Žádost o úvěr (1)

KLIENT	PŘÍJEM	KONTO	POHLAVÍ	NEZAMĚSTNANÝ	ÚVĚR
k1	VYSOKÝ	VYSOKÉ	ŽENA	NE	ANO
k2	VYSOKÝ	VYSOKÉ	MUŽ	NE	ANO
k3	NÍZKÝ	NÍZKÉ	MUŽ	NE	NE
k4	NÍZKÝ	VYSOKÉ	ŽENA	ANO	ANO
k5	NÍZKÝ	VYSOKÉ	MUŽ	ANO	ANO
k6	NÍZKÝ	NÍZKÉ	ŽENA	ANO	NE
k7	VYSOKÝ	NÍZKÉ	MUŽ	NE	ANO
k8	VYSOKÝ	NÍZKÉ	ŽENA	ANO	ANO
k9	NÍZKÝ	STŘEDNÍ	MUŽ	ANO	NE
k10	VYSOKÝ	STŘEDNÍ	ŽENA	NE	ANO
k11	NÍZKÝ	STŘEDNÍ	ŽENA	ANO	NE
k12	NÍZKÝ	STŘEDNÍ	MUŽ	NE	ANO

Rozhodovací stromy
Příklad: Žádost o úvěr (2)

Čtyřpolní tabulka pro příjem a úvěr:

	ÚVĚR ANO	ÚVĚR NE
PŘÍJEM VYSOKÝ	5	0
PŘÍJEM NÍZKÝ	3	4

Volba atributu pro větvení podle nejnižší entropie:

- **PŘÍJEM:**

$$H(PRIJEM) = \frac{5}{12} H(PRIJEM(VYSOKY)) + \frac{7}{12} H(PRIJEM(NIZKY)) =$$

$$= \frac{5}{12} \cdot 0 + \frac{7}{12} \cdot 0.9852 = 0.5747$$

Rozhodovací stromy
Příklad: Žádost o úvěr (3)

Volba atributu pro větvení podle nejnižší entropie:

- **PŘÍJEM** (pokračování):

$$H(PRIJEM\ (VYSOKY)) = -p_+ \log_2 p_+ - p_- \log_2 p_- =$$

$$= -\frac{5}{5}\log_2\frac{5}{5} - \frac{0}{5}\log_2\frac{0}{5} = 0 + 0 = 0$$

$$H(PRIJEM\ (NIZKY)) = -p_+ \log_2 p_+ - p_- \log_2 p_- =$$

$$= -\frac{3}{7}\log_2\frac{3}{7} - \frac{4}{7}\log_2\frac{4}{7} = 0.9852$$

Rozhodovací stromy
Příklad: Žádost o úvěr (4)

Volba atributu pro větvení podle nejnižší entropie:

- **KONTO:** $$H(KONTO) = \frac{4}{12} H(KONTO(VYSOKE)) + \frac{4}{12} H(KONTO(STREDNI)) + \frac{4}{12} H(KONTO(NIZKE)) =$$
$$= \frac{1}{3} \cdot 0 + \frac{1}{3} \cdot 1 + \frac{1}{3} \cdot 1 = 0.6667$$

Rozhodovací stromy
Příklad: Žádost o úvěr (5)

Volba atributu pro větvení podle nejnižší entropie:

- **POHLAVÍ:**

$$H(POHLAVI) = \frac{6}{12} H(POHLAVI(MUZ)) +$$

$$+ \frac{6}{12} H(POHLAVI(ZENA)) =$$

$$= \frac{1}{2} \cdot 0.9183 + \frac{1}{2} \cdot 0.9183 = 0.9183$$

Rozhodovací stromy
Příklad: Žádost o úvěr (6)

Volba atributu pro větvení podle nejnižší entropie:

- **NEZAMĚSTNANÝ:**

$$H(NEZAMESTNANY) = \frac{6}{12} H(NEZAMESTNANY(ANO)) +$$

$$+ \frac{6}{12} H(NEZAMESTNANY(NE)) =$$

$$= \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot 0.65 = 0.825$$

Rozhodovací stromy
Příklad: Žádost o úvěr (7)

Volba atributu pro větvení podle nejnižší entropie:

→ První větvení pro atribut **PRIJEM**: 2 třídy

- PRIJEM (VYSOKY): UVER (ANO) pro všechny vzory
- PRIJEM (NIZKY): 7 klientů + větvení
  - **KONTO:**

$$H(KONTO) = \frac{2}{7}H(KONTO(VYSOKE)) + \frac{3}{7}H(KONTO(STREDNI)) + \frac{2}{7}H(KONTO(NIZKE)) = 0.3935$$

  - **POHLAVÍ:** ***H (POHLAVI) = 0.965***
  - **NEZAMĚSTNANÝ:** ***H (NEZAMESTNANY) = 0.9792***

Rozhodovací stromy
Příklad: Žádost o úvěr (8)

Volba atributu pro větvení podle nejnižší entropie:

→ Druhé větvení podle atributu **KONTO**: 3 třídy

- KONTO (VYSOKE): UVER (ANO) pro všechny vzory
- KONTO (NIZKE): UVER (NE) pro všechny vzory
- KONTO (STREDNI): 3 klienti + větvení
  - **POHLAVÍ:**

$$H(POHLAVI) = \frac{2}{3} H(POHLAVI(MUZ)) + \frac{1}{3} H(POHLAVI(ZENA)) =$$

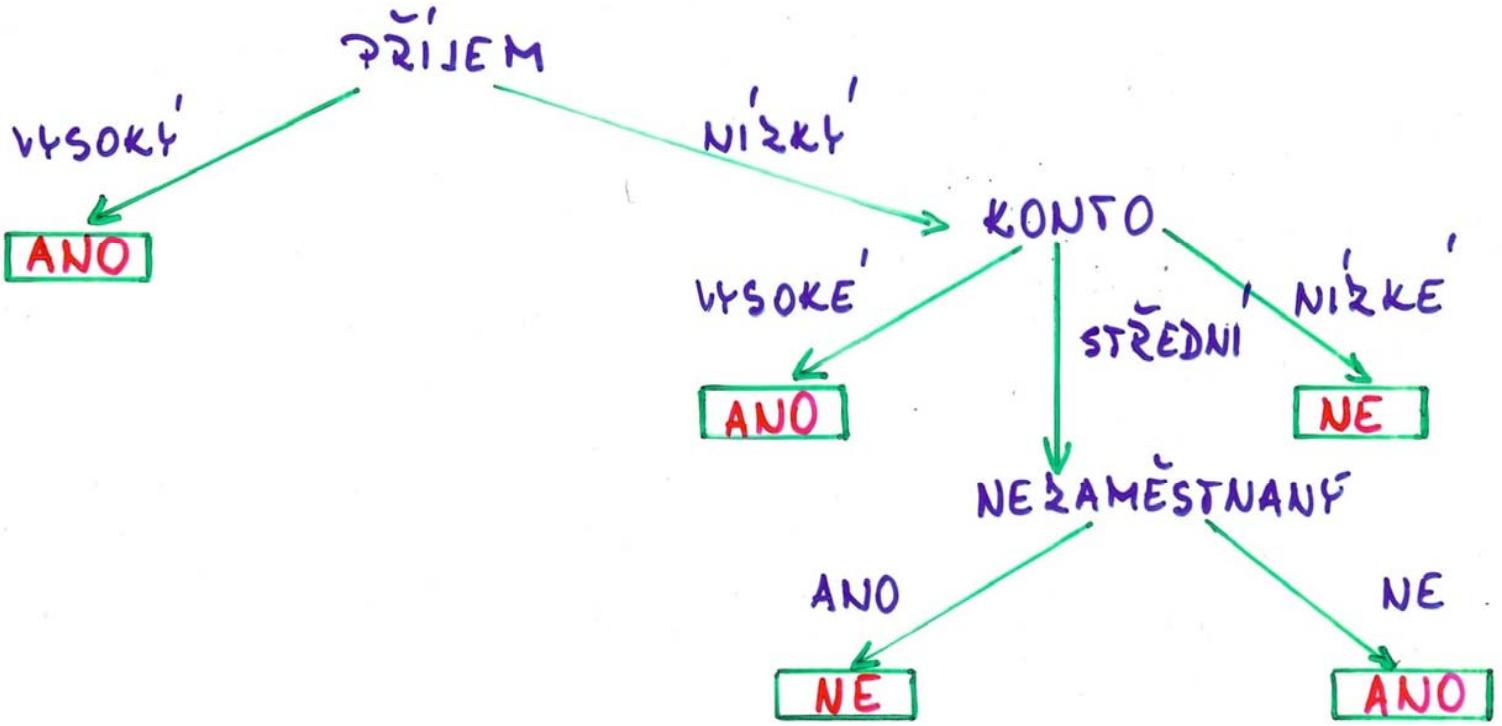
$$= \frac{2}{3} \cdot 1 + \frac{1}{3} \cdot 0 = 0.6667$$

  - **NEZAMĚSTNANÝ:** ***H (NEZAMESTNANY) = 0***

→ Třetí větvení podle atributu **NEZAMESTNANY**

Rozhodovací stromy
Příklad: Žádost o úvěr (9)

Indukovaný rozhodovací strom:



Rozhodovací stromy
Příklad: Žádost o úvěr (10)

Převod indukovaného stromu na pravidla:

IF *PRIJEM (VYSOKY)* **THEN** *UVER (ANO)*

IF *PRIJEM (NIZKY)* $\wedge$ *KONTO (VYSOKE)*
THEN *UVER (ANO)*

IF *PRIJEM (NIZKY)* $\wedge$ *KONTO (NIZKE)*
THEN *UVER (NE)*

IF *PRIJEM (NIZKY)* $\wedge$ *KONTO (STREDNI)* $\wedge$
NEZAMESTNANY (ANO) **THEN** *UVER (NE)*

IF *PRIJEM (NIZKY)* $\wedge$ *KONTO (STREDNI)* $\wedge$
NEZAMESTNANY (NE) **THEN** *UVER (ANO)*

Rozhodovací stromy (13)

Faktory důležité při indukci rozhodovacího stromu:

- **Volba atributů pro dělení**
  - Vliv na efektivitu vytvářeného stromu
  - ´informovaný´ vstup experta pro danou oblast
- **Uspořádání atributů pro dělení**
  - Omezit zbytečná porovnávání
- **Dělení**
  - Počet potřebných dělení
  - Obtížnější pro spojité hodnoty anebo velký počet hodnot

Rozhodovací stromy (14)

Faktory důležité při indukci rozhodovacího stromu:

- **Tvar vytvořeného stromu**
  - Vhodnější jsou vyvážené stromy s co nejmenším počtem úrovní
    x složitější porovnávání
  - Některé algoritmy vytvářejí jen binární stromy (často hlubší)
- **Ukončovací kritéria**
  - Správná klasifikace trénovacích dat
  - Předčasné ukončení zabraňuje vytváření velkých stromů a přeučení
    → **přesnost x efektivita**
  - **Occam´s razor** (William of Occam – 1320)
    preference nejjednodušší hypotézy pro D

Rozhodovací stromy (15)

Faktory důležité při indukci rozhodovacího stromu:

- **Trénovací data a jejich množství**
  - Málo → vytvořený strom nemusí být dostatečně spolehlivý pro obecnější data
  - Příliš mnoho → nebezpečí přeučení
- **Prořezávání**
  - Vyšší přesnost a efektivita pro testovaná data
  - Odstranit redundantní porovnávání, případně celé podstromy, resp. nepodstatné atributy
  - Přeučení → odstranění celých podstromů na nižších úrovních

Rozhodovací stromy (16)

Faktory důležité při indukci rozhodovacího stromu:

- **Prořezávání** (pokračování)
  - Lze provést již během indukce stromu
    → omezí vytváření příliš velkých stromů
  - Jiný přístup – prořezávání již vytvořených stromů
- **Časová a prostorová složitost**
  - Závisí na množství trénovacich dat ***q*** , počtu atributů ***h*** a tvaru indukovaného stromu – hloubka, větvení
  - Časová složitost pro indukci stromu: ***O (h q log q)***
  - Časová složitost klasifikace ***n*** vzorů stromem hloubky ***O (log q)*** : ***O (n log q)***

Rozhodovací stromy (17)

ID3 – Algoritmus (Ross Quinlan, 1986):

- Učení funkcí s Boolovskými hodnotami
- Greedy metoda
- Vytváří strom zeshora dolů
- V každém uzlu zvolí atribut, který nejlépe klasifikuje lokální trénovací data

  → nejlepší atribut má nejvyšší informační zisk
- Proces pokračuje, dokud nejsou všechna trénovací data správně zařazena, anebo dokud nebyly použity všechny atributy

Rozhodovací stromy (18)

ID3 – Algoritmus (pokračování):

- ***EXAMPLES*** ~ trénovací data (vzory)
- ***TARGET_ATTRIBUTE*** ~ atribut, jehož hodnotu má strom predikovat (třída)
- ***ATTRIBUTES*** ~ seznam atributů, které má vytvářený strom testovat
- Vrací rozhodovací strom, který správně klasifikuje daná trénovací data (***EXAMPLES***)
- **Entropie** ~ míra neuspořádanosti (~ „překvapení“) v trénovací množině
- **Informační zisk** ~ očekávaná redukce entropie po rozdělení dat podle uvažovaného atributu
- **Preference ´menších´stromů x přeučení**

Rozhodovací stromy (19)

ID3 – Algoritmus (pokračování):

ID3(EXAMPLES, TARGET_ATTRIBUTE, ATTRIBUTES)

- Vytvoř kořen ***ROOT*** stromu
- ***IF*** všechny ***EXAMPLES*** pozitivní, ***RETURN*** strom ***ROOT***, který má jediný uzel ohodnocený ´+´
- ***IF*** všechny ***EXAMPLES*** negativní, ***RETURN*** strom ***ROOT***, který má jediný uzel ohodnocený ´-´
- ***IF ATTRIBUTES = {}*** , ***RETURN*** strom ***ROOT***, který má jediný uzel ohodnocený nejčastější hodnotou ***TARGET_ATTRIBUTE*** v ***EXAMPLES***

Rozhodovací stromy (20)

// ***ID3(EXAMPLES, TARGET_ATTRIBUTE, ATTRIBUTES)***
// pokračování

- ***ELSE BEGIN***
  - ***A*** <= atribut z ***ATTRIBUTES*** , který nejlépe klasifikuje ***EXAMPLES***
  - Atribut pro dělení v ***ROOT*** <= ***A***
  - Pro všechny možné hodnoty v_i atributu ***A***
    - Připoj k ***ROOT*** novou hranu, která odpovídá testu $A = v_i$
    - Nechť $EXAMPLES v_i$ je podmnožina ***EXAMPLES*** s hodnotou v_i pro ***A***

Rozhodovací stromy (21)

// ***ID3(EXAMPLES, TARGET_ATTRIBUTE, ATTRIBUTES)***

// pokračování

- ***IF EXAMPLESv_i = {}***
  - Připoj k hraně list ohodnocený nejčastější hodnotou ***TARGET_ATTRIBUTE*** v ***EXAMPLES***
  - ***ELSE*** připoj k hraně podstrom ***ID3(EXAMPLESv_i , TARGET_ATTRIBUTE, ATTRIBUTES - {A})***
- ***END***
- ***RETURN ROOT***

Rozhodovací stromy (22)

ID3 – Algoritmus (pokračování):

- Kritérium pro dělení: **informační zisk**
  - výpočet pomocí **entropie**:
    - ***c*** … tříd
    - p_i … pravděpodobnost třídy ***i***
      (~ počet vzorů z ***i*** / počet všech vzorů v tréno-
      vací množině ***EXAMPLES***)

$$ENTROPY(EXAMPLES) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

Rozhodovací stromy (23)

ID3 – Algoritmus (pokračování):

- Kritérium pro dělení: **informační zisk**
  - **Informační zisk - GAIN**:
    - ***A*** … uvažovaný atribut
    - ***VALUES(A)*** … množina všech možných hodnot pro atribut ***A***

$$GAIN(EXAMPLES, A) = ENTROPY(EXAMPLES) -$$

$$- \sum_{v \in VALUES(A)} \frac{|EXAMPLES_v|}{|EXAMPLES|} ENTROPY(EXAMPLES_v)$$

Rozhodovací stromy (24)

Algoritmus C4.5 (Ross Quinlan - 1993):

- Modifikace algoritmu ID3
- Chybějící data
  - Při konstrukci stromu se ignorují
    → při výpočtu kritéria pro dělení se uvažují pouze známé hodnoty daného atributu
  - Při klasifikaci se chybějící hodnota atributu odhadne podle hodnot známých pro ostatní vzory
- Spojitá data
  - Kategorizace dat podle hodnot atributu na vzorech z trénovací množiny

Rozhodovací stromy (25)

Algoritmus C4.5 (pokračování):

- Prořezávání (pruning)
  - **Validace**: rozdělení trénovacích dat na trénovací množinu (2/3) a validační množinu (1/3)
  - **Náhrada podstromu** (reduced – error pruning) listem ohodnoceným nejčastější třídou na příslušných trénovacích vzorech
    - **Nový strom nesmí být na validační množině horší než původní!**
    - V opačném případě se náhrada podstromu neprovede

Rozhodovací stromy (25)

Algoritmus C4.5 (pokračování):

- Prořezávání (pruning)
  - **Roubování** (subtree raising)
    - Náhrada podstromu jeho nejčastěji užívaným podstromem – ten je tak přenesen na vyšší úroveň
    - Test na přesnost klasifikace
- Pravidla a jejich zjednodušení (rule post-pruning)
  - Inference rozhodovacího stromu z trénovací množiny – povoleno přeučení
  - Konverze vytvořeného stromu do ekvivalentní sady pravidel
    - Pro každou cestu od kořene k listu je vytvořeno jedno pravidlo

Rozhodovací stromy (26)

Algoritmus C4.5 (pokračování):

- Pravidla a jejich zjednodušení (rule post-pruning)
  - Prořezání (~ generalizace) pravidel odstraněním všech možných předpokladů, pokud to nepovede ke zhoršení odhadované správnosti klasifikace
  - Uspořádání prořezaných pravidel podle jejich očekávané přesnosti
    - V tomto pořadí se pravidla použijí při následné klasifikaci vzorů

→ snazší a radikálnější prořezávání než v případě celých rozhodovacích stromů

→ transparentní, srozumitelná pravidla

Rozhodovací stromy (27)

Algoritmus C4.5 (pokračování):

- Odhad správnosti pravidel, resp. stromů
  - Standardní přístup:
    - Použít validační množinu
  - **C4.5:**
    - **Výpočet správnosti na trénovacích datech**
      ~ Podíl správně klasifikovaných vzorů pokrytých pravidlem a všech příkladů pokrytých pravidlem
    - **Výpočet směrodatné odchylky správnosti**
      ~ Předpokládá se binomické rozdělení, kdy zjišťujeme pravděpodobnost, že na daném počtu vzorů dosáhneme daný počet správných rozhodnutí

Rozhodovací stromy (28)

Algoritmus C4.5 (pokračování):

- Odhad správnosti pravidel, resp. stromů

  → **Jako hledaná charakteristika pravidla se vezme dolní odhad správnosti pro zvolený interval spolehlivosti**

  - Příklad: Pro interval spolehlivosti 95% bude dolní odhad správnosti pro nová data:

    SPRÁVNOST_NA_TRÉNOVACÍCH_DATECH –
    – 1.96 x SMĚRODATNÁ_ODCHYLKA

Rozhodovací stromy (29)

Algoritmus C4.5 (pokračování):

- Kritérium pro dělení: **poměrný informační zisk**

$$GAIN\_RATIO(EXAMPLES, A) = \frac{GAIN(EXAMPLES, A)}{-\sum_{v \in VALUES(A)} \left(\frac{|EXAMPLES_v|}{|EXAMPLES|} \log_2 \frac{|EXAMPLES_v|}{|EXAMPLES|} \right)}$$

- Pro dělení se použije atribut s nejvyšším poměrným informačním ziskem

Rozhodovací stromy (30)

Algoritmus C5.0:

- Komerční verze C4.5 pro velké databáze
- Podobná indukce rozhodovacího stromu
- Efektivnější generování pravidel (zatím nezveřejněno)
- Vyšší dosahovaná spolehlivost:
  - **Boosting** (~ kombinace několika klasifikátorů)
    - Z trénovacích dat vytvoří několik trénovacích množin
    - Každému trénovacímu vzoru je přiřazena váha odpovídající jeho významu při klasifikaci
    - Pro každou kombinaci vah je vytvořen jiný klasifikátor
    - Při následné klasifikaci vzorů má každý klasifikátor jeden hlas, vítězí většina

Rozhodovací stromy (31)

Klasifikační a regresní stromy – algoritmus CART:

~ **C**lassification **A**nd **R**egression **T**rees

- Generuje binární rozhodovací stromy
- Výběr nejlepšího atributu pro dělení – entropie, Gini-index
- Při dělení se vytvoří jen dvě hrany
- Dělení probíhá podle ´nejlepšího´ bodu v daném uzlu ***t***
- Procházejí se všechny možné hodnoty ***s*** atributu pro dělení
  - ***L, R*** … levý, resp. pravý podstrom
  - P_L , P_R … pravděpodobnost zpracování příslušným podstromem

$$P_L = \frac{POCET\_VZORU\_V\_PODSTROMU\_L}{POCET\_VZORU\_V\_CELE\_TRENOVACI\_MNOZINE}$$

Rozhodovací stromy (32)

Klasifikační a regresní stromy (pokračování):

- Procházejí se všechny možné hodnoty ***s*** atributu pro dělení
  - V případě rovnosti se použije pravý podstrom
  - $P(C_j | t_L)$, $P(C_j | t_R)$ … pravděpodobnost, že vzor je v třídě C_j a v levém, resp. pravém podstromu

$$P(C_j | t) = \frac{POCET\_VZORU\_TRIDY\_j\_V\_PODSTROMU}{POCET\_VZORU\_V\_UVAZOVANEM\_UZLU}$$

  - Volba kritéria pro dělení:

$$\Phi(s | t) = 2 P_L P_R \sum_{j=1}^{c} \left| P(C_j | t_L) - P(C_j | t_R) \right|$$

Rozhodovací stromy (33)

Klasifikační a regresní stromy (pokračování):

- Vlastnosti CART:
  - Uspořádání atributů podle jejich významu při klasifikaci
  - Chybějící údaje ignoruje – nezapočítávají se při vyhodnocování kritéria pro dělení
  - Ukončovací kritérium:
    - Žádné další dělení by nezvýšilo přesnost klasifikace
  - Vysoká přesnost na trénovací množině nemusí odpovídat přesnosti na testovaných datech

Rozhodovací stromy (34)

Algoritmus CHAID:

~ **Ch**i-square **A**utomatic **I**nteraction **D**etection

- Kritérium pro větvení: χ^2
- Algoritmus automaticky seskupuje hodnoty kategoriálních atributů
  - Hodnoty atributu se postupně seskupují z původního počtu až do dvou skupin
  - Následně se vybere atribut a jeho kategorizace, která je v daném kroku pro větvení nejlepší

→ při větvení se nevytváří tolik větví, kolik má atribut hodnot

Rozhodovací stromy (35)

Algoritmus CHAID - seskupování hodnot atributu:

1. Opakuj, dokud nevzniknou pouze dvě skupiny hodnot atributu
   1. Zvol dvojici kategorií atributu, které jsou si nejpodobnější z hlediska χ^2 a které mohou být spojeny
   2. Považuj novou kategorizaci atributu za možné shlukování v daném kroku
2. Pro každý z možných způsobů shlukování hodnot spočítej pomocí χ^2 –testu pravděpodobnost ***p***
3. Shlukování s nejnižší pravděpodobností ***p*** zvol za ´nejlepší´ shlukování hodnot atributu
4. Zjisti, zda toto nejlepší shlukování statisticky významně přispěje k odlišení příkladů z různých tříd

Rozhodovací stromy (36)

Algoritmus SPRINT:

~ **S**calable **Pa**ra**R**allelizable **IN**duction of decision **T**rees

- Použití pro velké databáze
  - Nepotřebuje uchovávat jednotlivé vzory v paměti
- Kritérium pro dělení: **Gini-index**
  - D … databáze (rozdělená na D_1 a D_2)
  - p_j … frekvence třídy C_j v D

$$GINI(D) = 1 - \sum_{j=1}^{c} p_j^2$$

Rozhodovací stromy (37)

Algoritmus SPRINT (pokračování):

- Kritérium pro dělení: **Gini-index**
  - ***n, n*$_1$*, n*$_2$ …** počet prvků ***D, D*$_1$*, D*$_2$**

$$GINI_{SPLIT}(D) = \frac{n_1}{n}\left(GINI(D_1)\right) + \frac{n_2}{n}\left(GINI(D_2)\right)$$

- Spojitá data – dělení uprostřed mezi dvěma po sobě jdoucími hodnotami
- Volba atributu pro dělení: metodou **Rainforest**

Rozhodovací stromy (38)

Algoritmus SPRINT - metoda Rainforest:

- Agregovaná metadata (vytvořená z atributů) se uchovávají v tabulce **AVC** ~ **A**ttribute – **V**alue **C**lass label group
- AVC-tabulka se vytvoří pro každý uzel rozhodovacího stromu
- Sumarizuje informaci potřebnou k určení atributů pro dělení
- Velikost tabulky závisí na počtu tříd, hodnot atributů a případných atributech pro dělení

  → Redukce dimenze pro velké trénovací množiny

Rozhodovací stromy (39)

Algoritmus SPRINT (pokračování):

- **Indukce rozhodovacího stromu:**
  - **Projdou se trénovací data**
  - **Vytvoří se AVC-tabulka**
  - **Určí se nejlepší atribut pro dělení**
  - **Rozdělení trénovacích dat**
  - **Konstrukce AVC-tabulky pro následující uzel**