5.1 Rozhodovací stromy

5.1.1 Základní algoritmus

Způsob reprezentování znalostí v podobě rozhodovacích stromů je dobře znám z řady oblastí. Vzpomeňme jen na nejrůznější „klíče k určování" různých živočichů nebo rostlin známých z biologie. Indukce rozhodovacích stromů patří k nejznámějším algoritmům z oblasti symbolických metod strojového učení. Při tvorbě rozhodovacího stromu se postupuje metodou *rozděl a panuj* (divide and concquer). Trénovací data se postupně rozdělují na menší a menší podmnožiny (uzly stromu) tak, aby v těchto podmnožinách převládaly příklady jedné třídy. Na počátku tvoří celá trénovací data jednu množinu, na konci máme podmnožiny tvořené příklady téže třídy (Quinlan, 1979). Tento postup bývá často nazýván *top down induction of decision trees* (TDIDT). Postupuje se tedy metodou specializace v prostoru hypotéz (stromů) shora dolů, počínaje stromem s jedním uzlem (kořenem). Cílem je nalézt nějaký strom konzistentní s trénovacími daty[29], přitom se dává přednost menším, jednodušším stromům. Obecné schéma algoritmu pro tvorbu rozhodovacích stromů je na obr. 45.

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, pro tento uzel opakuj postup od bodu 1, jinak skonči.

Obr. 45 Obecný algoritmus pro tvorbu rozhodovacích stromů.

Uvedený algoritmus bude fungovat pro kategoriální data (počet podmnožin-uzlů vytvářený v kroku 2 odpovídá počtu hodnot daného atributu), která nejsou zatížena šumem (růst stromu se podle bodu 3 zastaví v okamžiku, kdy všechny příklady v daném uzlu patří do téže třídy). V dalších částech této kapitoly uvidíme, jak se dají tato dvě omezení překonat. Nejprve se však zaměříme na

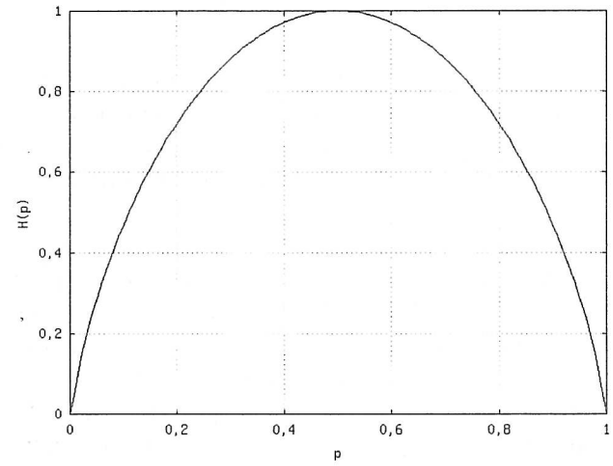
[29] Připomeňme, že algoritmus *Candidate-Elimination* hledal všechny takové hypotézy.

klíčovou otázku celého algoritmu; jak vybrat vhodný atribut pro větvení stromu (bod 1). Cíl je zřejmý: vybrat takový atribut, který od sebe nejlépe odliší příklady různých tříd. Vodítkem pro volbu jsou charakteristiky atributu převzaté z teorie informace nebo pravděpodobnosti: *entropie*, *informační zisk*, *poměrný informační zisk*, χ^2 nebo[30] *Gini index*.

Entropie je pojem používaný v přírodních vědách (např. fyzika) pro vyjádření míry neuspořádanosti nějakého systému. V teorii informace je entropie definovaná jako funkce

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

kde p_t je pravděpodobnost výskytu třídy t (v našem případě relativní četnost třídy t počítaná na určité množině příkladů) a T je počet tříd. Podobu funkce H v případě dvou tříd ukazuje obr. 46. Graf znázorňuje průběh entropie v závislosti na pravděpodobnosti p jedné ze tříd. Je-li $p = 1$ (všechny příklady patří do této třídy) nebo $p = 0$ (žádný příklad nepatří do této třídy), je entropie nulová. Jsou-li obě třídy zastoupeny stejným počtem příkladů ($p = 0{,}5$), je entropie maximální.



Obr. 46 Entropie.

Výpočet entropie pro jeden atribut se provádí tímto způsobem: Pro každou hodnotu v, kterou může nabýt uvažovaný atribut A spočítej podle uvedeného vzorce entropii $H(A(v))$ na skupině příkladů, které jsou pokryty kategorií $A(v)$

[30] Jsou ale i další možnosti. Například Mantaras (Mantaras, 1991) používá pro výběr atributu vzdálenost mezi atributem a třídou.

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

Spočítej střední entropii $H(\boldsymbol{A})$ jako vážený součet entropií $H((A(v))$, přičemž váhy v součtu jsou relativní četnosti kategorií $A(v)$ v datech $\mathbf{D}_{TR}$

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

Pro větvení stromu pak vybereme atribut s nejmenší entropií $H(\boldsymbol{A})$.

Informační zisk (information gain) i poměrný informační zisk (information gain ratio) jsou míry odvozené z entropie. Informační zisk se spočítá jako rozdíl entropie pro celá data (pro cílový atribut) a pro uvažovaný atribut. Informační zisk tak měří redukci entropie způsobenou volbou atributu $\boldsymbol{A}$:

$$Zisk(\boldsymbol{A}) = H(\boldsymbol{C}) - H(\boldsymbol{A}),$$

kde

$$H(\boldsymbol{C}) = -\sum_{t=1}^{T} \frac{n_t}{n} \log_2 \frac{n_t}{n}.$$

Zatímco v případě entropie jsme hledali atribut s minimální hodnotou, v případě informačního zisku hledáme atribut s maximální hodnotou. To je logické, uvážíme-li, že entropie počítaná pro celá data nezávisí na atributu. První člen rozdílu je tedy konstantní, takže rozdíl bude maximální v případě, že druhý člen rozdílu bude minimální.

Uvedená dvě kritéria mají jednu nevýhodu, protože neberou do úvahy počet hodnot zvoleného atributu. Důležité je pouze to, jak dobře tento atribut od sebe odliší příklady různých tříd. Pokud bychom jako atribut pro větvení vybrali například pořadové číslo příkladu, dosáhneme nejnižší (nulovou) entropii, protože jedné hodnotě atributu odpovídá jediný objekt. Tento atribut by nám tedy umožnil bezchybně klasifikovat trénovací data (tak, že bychom si „pamatovali", který objekt patří do které třídy), byl by ale zcela nepoužitelný pro klasifikaci nových příkladů. Proto se někdy používá jako kritérium pro volbu atributu *poměrný informační zisk*, který kromě entropie bere do úvahy i počet hodnot atributu

$$Poměrný\ zisk(\boldsymbol{A}) = \frac{Zisk(\boldsymbol{A})}{Větvení(\boldsymbol{A})}.$$

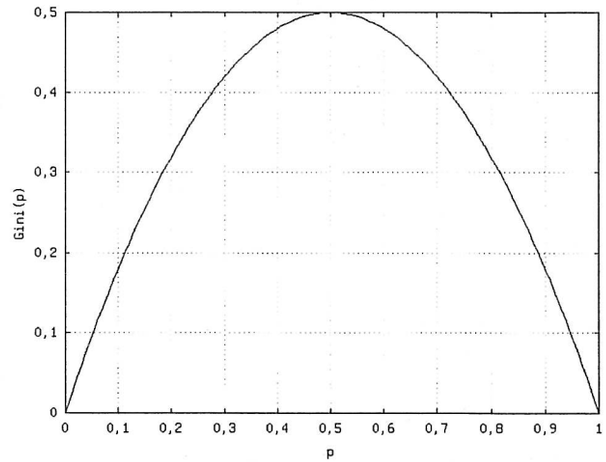
Jmenovatelem v uvedeném vztahu je vlastně entropie dat vzhledem k hodnotám atributu[31] $\boldsymbol{A}$:

$$Větvení(\boldsymbol{A}) = -\sum_{v \in Val(\boldsymbol{A})} \left(\frac{n(A(v))}{n} \log_2 \frac{n(A(v))}{n} \right).$$

Stejnou roli, jakou hrála v předcházejících úvahách entropie, může mít i Gini index. Tento index se spočítá jako

$$Gini = 1 - \sum_{t=1}^{T} (p_t^2),$$

kde p_t je opět relativní počet příkladů t-té třídy zjišťovaný na nějaké (pod)množině. Graf závislosti Gini indexu na pravděpodobnosti jedné ze dvou tříd ukazuje obr. 47. Opět je hodnota indexu minimální v případě, že příklady patří do jedné ze tříd, a maximální v případě, že příklady jsou rovnoměrně rozděleny mezi obě třídy.



Obr. 47 Gini index.

Hodnotu Gini indexu pro jeden atribut spočítáme analogicky jako hodnotu entropie pro jeden atribut. Tedy tak, že pro každý atribut spočítáme vážený součet indexu pro jednotlivé hodnoty atributu, přičemž váhy budou opět relativní četnosti příslušných hodnot

[31] V předcházejících úvahách jsme pracovali s entropií dat vzhledem k cílovým třídám. Zde je tedy entropie použita v trochu jiném smyslu.

$$Gini(\boldsymbol{A}) = \sum_{v \in Val(A)} \frac{n(A(v))}{n} Gini(A(v)), \quad Gini(A(v)) = 1 - \sum_{t=1}^{T} \left(\frac{n_t(A(v))}{n(A(v))} \right)^2 .$$

Pro větvení použijeme atribut, který bude mít nejmenší hodnotu tohoto indexu. Můžeme samozřejmě maximalizovat i rozdíl mezi Gini indexem počítaným pro cílový atribut a Gini indexem jednoho atributu (analogie s informačním ziskem)

$$Gini(\boldsymbol{C}) - Gini(\boldsymbol{A}),$$

kde

$$Gini(\boldsymbol{C}) = 1 - \sum_{t=1}^{T} \left(\frac{n_t}{n} \right)^2 .$$

Jako kritérium pro volbu atributu je možné použít i χ^2. Tato míra, umožňující vyhodnocovat vzájemnou souvislost mezi dvěma atributy, je podrobněji popsána v kapitole věnované statistickým metodám. Při volbě vhodného atributu pro větvení stromu se postupuje tak, že se vybere atribut, který nejvíce souvisí s cílovým atributem (χ^2 má největší hodnotu).

Tabulka 12 Příklad dat pro tvorbu stromu

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*

Celý postup indukce rozhodovacího stromu snad lépe osvětlí následující numerický příklad. Vezměme data uvedená v tab. 12. Na počátku jsou všechny trénovací příklady v jedné množině. Volbu atributu pro první větvení budeme tedy vybírat na základě všech 12 příkladů tak, že spočítáme entropii pro jednotlivé atributy. Budeme zjišťovat, jak jsou jednotlivé hodnoty atributu rozděleny mezi příklady obou tříd.

Tabulka 13 Čtyřpolní tabulka pro příjem a úvěr

	úvěr ano	*úvěr ne*
příjem vysoký	5	0
příjem nízký	3	4

Entropii pro atribut *příjem* spočítáme z údajů uvedených v tab. 13, tedy

$$H(příjem) = \frac{5}{12} H(příjem(vysoký)) + \frac{7}{12} H(příjem(nízký)),$$

přičemž

$$H(příjem(vysoký)) = -p_+ \log_2 p_+ - p_- \log_2 p_- = -\frac{5}{5}\log_2 \frac{5}{5} - \frac{0}{5}\log_2 \frac{0}{5}$$
$$= 0 + 0 = 0,$$

$$H(příjem(nízký)) = -p_+ \log_2 p_+ - p_- \log_2 p_- = -\frac{3}{7}\log_2 \frac{3}{7} - \frac{4}{7}\log_2 \frac{4}{7}$$
$$= 0{,}9852,$$

tedy

$$H(příjem) = \frac{5}{12} 0 + \frac{7}{12} 0{,}9852 = 0{,}5747 .$$

Podobně spočítáme entropie pro další atributy

$$H(konto) = \frac{4}{12} H(konto(vysoké)) + \frac{4}{12} H(konto(střední)) + \frac{4}{12} H(konto(nízké))$$
$$= \frac{1}{3} 0 + \frac{1}{3} 1 + \frac{1}{3} 1 = 0{,}6667,$$

$$H(pohlaví) = \frac{6}{12} H(pohlaví(muž)) + \frac{6}{12} H(pohlaví(žena))$$
$$= \frac{1}{2} 0{,}9183 + \frac{1}{2} 0{,}9183 = 0{,}9183,$$

$$H(nezaměstnaný) = \frac{6}{12} H(nezaměstnaný(ano)) + \frac{6}{12} H(nezaměstnaný(ne))$$
$$= \frac{1}{2} 1 + \frac{1}{2} 0{,}6500 = 0{,}8250.$$

Pro větvení tedy vybereme atribut *příjem*. Dostaneme tak dvě podmnožiny trénovacích dat: příklady pokryté kategorií *příjem*(*vysoký*) patří všechny do třídy *úvěr*(*ano*), příklady pokryté kategorií *příjem*(*nízký*) patří do různých tříd. V následujícím kroku algoritmu TDIDT tedy budeme hledat atribut, který od sebe oddělí příklady s nízkým příjmem. Pro atributy *konto*, *pohlaví* a *nezaměstnaný* tedy budeme opět počítat entropii, tentokrát ale již jen na skupině 7 příkladů, klientů s nízkým příjmem:

$$H(konto) = \frac{2}{7}H(konto(vysoké)) + \frac{3}{7}H(konto(střední)) + \frac{2}{7}H(konto(nízké))$$
$$= \frac{2}{7}0 + \frac{3}{7}0{,}9183 + \frac{2}{7}0 = 0{,}3935,$$

$$H(pohlaví) = \frac{4}{7}H(pohlaví(muž)) + \frac{3}{7}H(pohlaví(žena))$$
$$= \frac{4}{7}1 + \frac{3}{7}0{,}9183 = 0{,}9650,$$

$$H(nezaměstnaný) = \frac{5}{7}H(nezaměstnaný(ano)) + \frac{2}{7}H(nezaměstnaný(ne))$$
$$= \frac{5}{7}0{,}9709 + \frac{2}{7}1 = 0{,}9792.$$

Klienty s nízkým příjmem tedy budeme větvit na základě výše konta. Příklady pokryté kategorií *konto(vysoké)* patří do třídy *úvěr(ano)*, příklady pokryté kategorií *konto(nízké)* patří do třídy *úvěr(ne)*. Zbývají příklady pokryté kategoriemi *konto(střední)* nebo *příjem(nízký)*, které patří do různých tříd. Budeme tedy pokračovat ve větvení podle jednoho z atributů *pohlaví*, *nezaměstnaný*. Opět spočítáme entropii, nyní pro tři příklady pokryté kombinací *konto(střední)* ∧ *příjem(nízký)* :

$$H(pohlaví) = \frac{2}{3}H(pohlaví(muž)) + \frac{1}{3}H(pohlaví(žena))$$
$$= \frac{2}{3}1 + \frac{1}{3}0 = 0{,}6667,$$

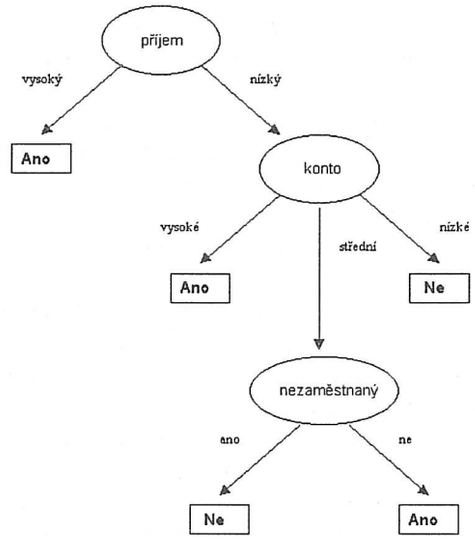
$$H(nezaměstnaný) = \frac{2}{3}H(nezaměstnaný(ano)) + \frac{1}{3}H(nezaměstnaný(ne))$$
$$= \frac{2}{3}0 + \frac{1}{3}0 = 0.$$

Vybereme atribut *nezaměstnaný*, který bezchybně rozdělí zbylé tři příklady. Výsledný strom je na obrázku 48. V nelistových uzlech stromu jsou uvedeny atributy použité při větvení, hrany stromu odpovídají hodnotám těchto atributů, a v listech stromu je informace o přiřazení ke třídě.

Použití rozhodovacího stromu pro klasifikaci nových případů je velmi prosté. Počínaje kořenem stromu se postupně zjišťují hodnoty atributů. Konkrétní hodnota odpovídá určité větvi, která nás přivede k dalšímu atributu atd., až se dostaneme do listového uzlu, který odpovídá třídě, do níž máme nový příklad zařadit. Tedy například klient s charakteristikami

příjem(nízký), *konto(nízké)*, *pohlaví(muž)*, *nezaměstnaný(ano)*

bude na základě větve, která je na obrázku 48 úplně vpravo, zařazen do třídy *úvěr(ne)*. Povšimněme si, že uvedený klient nebyl součástí trénovacích dat. Náš strom má tedy schopnost generalizovat.



Obr. 48 Úplný rozhodovací strom.

5.1.2 Převod stromu na pravidla

Uvedený příklad použití rozhodovacího stromu naznačuje, jak lze převést rozhodovací strom na sadu rozhodovacích pravidel. Každé cestě stromem od kořene k listu odpovídá jedno pravidlo. Nelistové uzly (atributy) se (spolu s hodnotou pro příslušnou hranu) objeví v předpokladu pravidla, listový uzel (cíl) bude v závěru pravidla. Rozhodovací strom z obrázku 48 lze tedy přepsat na pravidla uvedená na obr. 49. Převádění stromu na pravidla zvyšuje srozumitelnost nalezených znalostí, může být i vhodnější pro automatizované použití v nějakém klasifikačním systému (seznam pravidel se snadněji kóduje a strojově interpretuje).

```
IF příjem(vysoký)   THEN úvěr(ano)
IF příjem(nízký) ∧ konto(vysoké)   THEN úvěr(ano)
IF příjem(nízký) ∧ konto(střední) ∧ nezaměstnaný(ano)   THEN úvěr(ne)
IF příjem(nízký) ∧ konto(střední) ∧ nezaměstnaný(ne)    THEN úvěr(ano)
IF příjem(nízký) ∧ konto(nízké)    THEN úvěr(ne)
```

Obr. 49 Strom převedený na pravidla.

5.1.3 Prořezávání stromů

Rozhodovací strom z obr. 48 bezchybně klasifikuje všechny trénovací příklady uvedené v tabulce 12. Postupuje tedy důsledně podle algoritmu TDIDT; větvení skončí až ve chvíli, kdy všechny příklady odpovídající jednotlivým listovým uzlům patří do téže třídy (krok 3 algoritmu). Někdy však tento postup není ani žádoucí, ani možný. Požadavek na bezchybnou klasifikaci trénovacích dat může vést k přeučení (overfitting). Navíc bývá výsledný strom příliš košatý, a tedy málo srozumitelný. Bezchybná klasifikace trénovacích dat nebývá možná v situacích, kdy jsou trénovací data zatížena šumem. Proto se v realizovaných implementacích algoritmu požaduje, aby v listovém uzlu „převažovaly" příklady jedné třídy. Vedlejším efektem této změny je skutečnost, že výsledný strom bývá menší, a tedy srozumitelnější pro interpretaci (ovšem za cenu zhoršeného chování při klasifikaci trénovacích dat). K tomuto „redukovanému" stromu se lze propracovat dvěma způsoby:

- modifikací původního algoritmu (redukovaný strom se vytvoří přímo) nebo
- *následným prořezáním* (post-pruning) úplného stromu.

V praktických úlohách se jako úspěšnější osvědčil druhý způsob (je totiž poměrně obtížné poznat, kdy předčasně ukončit růst stromu). Zde se nejprve (i za cenu přeučení) vytvoří úplný strom. Ve fázi prořezávání se pak pro jednotlivé nelistové uzly posuzuje, do jaké míry úplný strom zhorší náhrada tohoto uzlu (a tedy i odpovídajícího podstromu) listem. Náhrada nelistového uzlu listem totiž znamená, že všechny příklady v tomto uzlu, budou zařazeny do téže třídy.

Při vytváření redukovaného stromu (ať už prvním, nebo druhým způsobem) je klíčovou otázkou jak poznat, kdy lze nelistový uzel nahradit listem. K rozhodnutí lze použít buď nová data (tzv. validační), která se použijí pro testování uvažované redukce, nebo se vhodnost redukce odhaduje na základě statistického testu pouze z trénovacích dat. Jako příklad prořezávání zmiňme prořezávání založené na pravidlech z obr. 50. Uvedený algoritmus odhaduje vhodnost prořezávání na základě trénovacích dat. Používá se při tom pesimistický odhad toho, jak se bude pravidlo chovat při klasifikaci dat neznámých[32]:

1. Spočítej správnost pravidla na trénovacích datech jako podíl správně klasifikovaných příkladů pokrytých pravidlem a všech příkladů pokrytých pravidlem,
2. Spočítej směrodatnou odchylku této správnosti (za předpokladu binomického rozdělení, kdy zjišťujeme pravděpodobnost, že na daném počtu příkladů dosáhneme daný počet správných rozhodnutí),
3. Vezmi dolní odhad správnosti pro zvolený interval spolehlivosti jako hledanou charakteristiku pravidla.

[32] Tento postup lze použít i pro odhad přesnosti celého stromu (modelu).

Tedy například pro interval spolehlivosti 95 % bude dolní odhad správnosti pravidla pro nová data

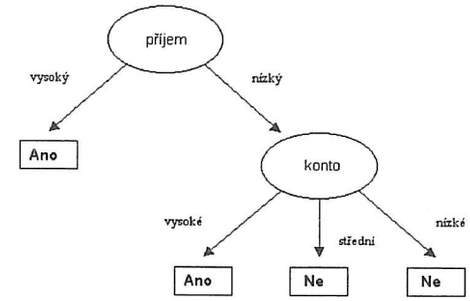
správnost na trénovacích datech – 1,96 *směrodatná odchylka.*

Použijeme-li uvedený postup na strom z obr. 48, získáme strom znázorněný na obr. 51. Na rozdíl od úplného stromu, který klasifikoval trénovací příklady bezchybně, se nyní dopustíme jedné chyby.

Algoritmus prořezávání

1. převeď strom na pravidla,
2. generalizuj pravidlo odstraněním podmínky z předpokladu, pokud dojde ke zlepšení odhadované správnosti,
3. uspořádej prořezaná pravidla podle odhadované správnosti; v tomto pořadí budou pravidla použita pro klasifikaci

Obr. 50 Algoritmus prořezávání rozhodovacího stromu.



Obr. 51 Prořezaný rozhodovací strom.

5.1.4 Numerické atributy

V případě numerických atributů musíme řešit problém s velkým počtem možných hodnot. Nelze tedy pro každou hodnotu vytvořit samostatnou větev. Pomocí se jeví rozdělení oboru hodnot na intervaly. Tyto intervaly pak považujeme za diskrétní hodnoty atributu[33]. Problém práce s numerickými atributy se obvykle řeší v kroku předzpracování tak, jak je uvedeno oddílu 7.6. Systémy pro tvorbu rozhodovacích stromů mívají ale metody diskretizace přímo zabudovány v sobě.

V nejjednodušším případě se provádí rozdělení na dva intervaly, tzv. *binarizace*. Využívá se informace o tom, do které třídy patří příklad s konkrétní

[33] Podobný problém nastane i u kategoriálních atributů, které nabývají velkého počtu hodnot (např. poštovní směrovací čísla, nebo kódy profesí). V takovýchto situacích lze postupovat analogicky, tedy vytvářet skupiny hodnot. Není-li atribut ordinální, nelze ale již využít přirozeného uspořádání.

hodnotou diskretizovaného atributu; svou roli tedy opět může sehrát entropie. Při hledání dělicího bodu (cut-point), který rozdělí hodnoty do dvou intervalů, se postupuje následujícím způsobem:

1. Seřadíme vzestupně hodnoty diskretizovaného atributu ***A***,
2. Pro každé možnou hodnotu dělicího bodu[34] θ spočítáme střední entropii atributu

$$H(A_\theta) = \frac{n(A(<\theta))}{n} H(A(<\theta)) + \frac{n(A(>\theta))}{n} H(A(>\theta)).$$

První člen součtu v uvedeném vztahu se týká příkladů, které mají hodnotu atributu menší než θ ; $H(A(<\theta)$ je entropie na těchto příkladech, $n(A(<\theta))/n$ je relativní četnost těchto příkladů. Druhý člen součtu se analogicky týká příkladů, které mají hodnotu atributu větší než θ.

3. Vybereme dělicí bod, který dá nejmenší entropii.

Uvedená binarizace se provádí *on-line* v průběhu vytváření stromu. V kroku 2 algoritmu TDIDT se tedy berou do úvahy jak atributy kategoriální, tak numerické, u numerických se ale nejprve hledá vhodný práh pro binarizaci. Na rozdíl od kategoriálních atributů se může v jedné větvi stromu opakovat test na tentýž numerický atribut.

Opět se podíváme na jednoduchý numerický příklad. Pro data uvedená v tab. 14 a atribut *konto* se nejprve uvažuje $\theta = 22\,500$. Tomuto prahu odpovídá rozdělení příkladů do tříd podle tabulky 15. Z této tabulky spočítáme entropii

$$H(konto_{22500}) = \frac{3}{12} H(konto(<22500)) + \frac{9}{12} H(konto(>22500))$$
$$= \frac{1}{4} 0,9183 + \frac{3}{4} 0,5640 = 0,6526.$$

Tabulka 14 Numerická data

klient	*příjem*	*konto*	*úvěr*
K101	3000	15000	*ne*
K102	10000	15000	*ne*
K103	17000	15000	*ano*
K104	5000	30000	*ne*
K105	15000	30000	*ano*
K106	20000	50000	*ano*
K107	2000	60000	*ne*
K108	5000	90000	*ano*
K109	10000	90000	*ano*
K110	20000	90000	*ano*
K111	10000	100000	*ano*
K112	17000	100000	*ano*

[34] Postupně se zkouší vložit dělicí bod θ doprostřed mezi každé dvě po sobě následující hodnoty atributu.

Tabulka 15 Čtyřpolní tabulka pro $konto_{22500}$ a úvěr

	úvěr ano	*úvěr ne*
konto<22500	1	2
konto>22500	7	2

Pro následující dělicí bod θ =40 000 spočítáme entropii

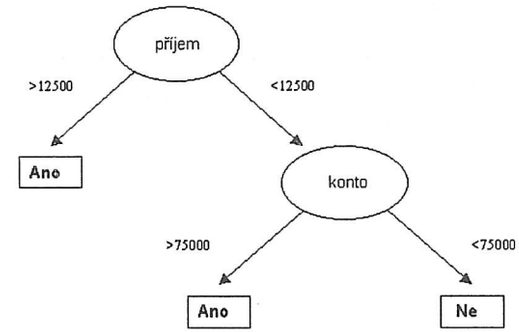
$$H(konto_{40000}) = \frac{5}{12} H(konto(<40000)) + \frac{7}{12} H(konto(>40000))$$
$$= \frac{5}{12} 0,9706 + \frac{7}{12} 0,5917 = 0,7497.$$

Podobně platí

$$H(konto_{55000}) = \frac{6}{12} H(konto(<55000)) + \frac{6}{12} H(konto(>55000))$$
$$= \frac{1}{2} 1 + \frac{1}{2} 0,6500 = 0,8250,$$

$$H(konto_{75000}) = \frac{7}{12} H(konto(<75000)) + \frac{5}{12} H(konto(>75000))$$
$$= \frac{7}{12} 0,9852 + \frac{5}{2} 0 = 0,5747,$$

$$H(konto_{95000}) = \frac{10}{12} H(konto(<95000)) + \frac{2}{12} H(konto(>95000))$$
$$= \frac{10}{12} 0,9709 + \frac{2}{12} 0 = 0,8091.$$



Obr. 52 Rozhodovací strom pro numerická data.

Z těchto výpočtů plyne, že nejvhodnější rozdělení hodnot atributu *konto* je dělicím bodem $\theta = 75\,000$ (entropie $H = 0{,}5747$). Podobně určíme pro atribut *příjem* nejvhodnější dělicí bod $\theta = 12\,500$ s entropií rovněž 0,5747. Pro kořen rozhodovacího stromu zvolíme atribut *příjem* (je první v seznamu atributů). Větev *příjem* > 12 500 bezchybně zařadí 5 klientů, kterým lze poskytnout úvěr, ve větvi *příjem* < 12 500 musíme najít vhodný práh atributu *konto*, který umožní odlišit zbylých 7 příkladů z trénovací množiny (obr. 52).

5.1.5 Chybějící hodnoty

V reálných úlohách dobývání znalostí se můžeme dostat do situace, kdy nám budou chybět údaje o některých objektech. Problém práce s chybějícími hodnotami lze opět řešit v kroku předzpracování. Některé implementace algoritmů pro tvorbu rozhodovacích stromů se ale dokáží s tímto problémem vyrovnat přímo.

Jednou z možností je uvažovat místo chybějící hodnoty nějakého atributu nejčastější hodnotu tohoto atributu. Jinou možností je vzít do úvahy relativní četnosti všech hodnot tohoto atributu na trénovacích datech a místo chybějící hodnoty uvažovat všechny hodnoty s váhami danými relativními četnostmi. Jestliže tedy atribut A měl sedmkrát hodnotu x a třikrát hodnotu y, budeme místo chybějící hodnoty předpokládat hodnotu x s pravděpodobností 0,7 a hodnotu y s pravděpodobností 0,3. Místo jednoho příkladu s chybějící hodnotou tak dostaneme dva „částečné" příklady, z nichž každý bude procházet rozhodovacím stromem po jiné větvi[35]; oba začínají v uzlu, který odpovídá atributu A.

5.1.6 Ceny atributů

V řadě aplikací může hrát roli i cena za získání hodnoty nějakého atributu[36]. V medicínské aplikaci můžeme například požadovat změření teploty pacienta i jeho vyšetření na počítačovém tomografu.

Informace o takovéto ceně se může brát do úvahy již při vytváření rozhodovacího stromu. Například kritérium informačního zisku pro výběr atributu pro větvení lze modifikovat takto:

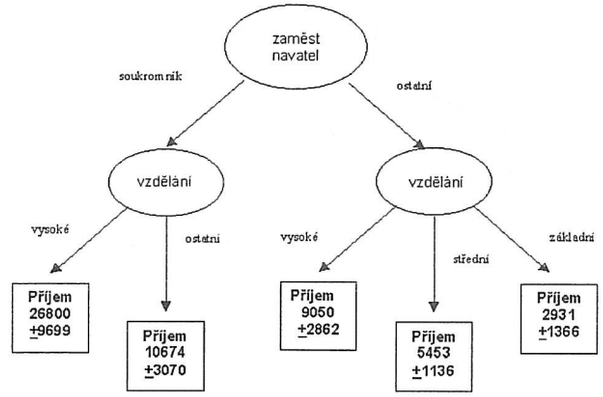
$$\frac{Zisk(\boldsymbol{A})^2}{Cena(\boldsymbol{A})},$$

kde $Cena(\boldsymbol{A})$ jsou náklady za zjištění hodnoty atributu.

[35] Tento způsob je použit například v Quinlanově systému C4.5 (Quinlan, 1993).
[36] Pozor, nezaměňovat tuto cenu s cenou za chybné rozhodnutí.

5.1.7 Regresní stromy

Zatím jsme předpokládali, že vytváříme stromy pro klasifikaci objektů do tříd. Takovým stromům se obvykle říká *klasifikační stromy*. Existují ale i *stromy regresní*, které umožňují odhadovat hodnotu nějakého numerického atributu. V listových uzlech mají takové stromy místo názvu třídy například konkrétní hodnotu (konstantu), která odpovídá průměrné hodnotě cílového atributu pro příklady v tomto uzlu (obr. 53).



Obr. 53 Regresní strom.

Algoritmus pro tvorbu regresního stromu odpovídá algoritmu TDIDT. Rozdíl je ve způsobu volby atributu pro větvení. Místo entropie se vychází ze směrodatné odchylky hodnot cílového atributu. Tedy místo kritéria informačního zisku můžeme uvažovat kritérium *redukce směrodatné odchylky*:

$$S_y - \sum_{v \in Val(\boldsymbol{A})} \frac{n(A(v))}{n} S_y(A(v)),$$

kde S_y^2 je rozptyl hodnot cílového atributu pro celá trénovací data a $S_y^2(A(v))$ značí rozptyl[37] hodnot cílového atributu pro příklady pokryté kategorií $A(v)$.

Pro větvení vybereme atribut, který maximalizuje toto kritérium. Větvení skončí, pokud se hodnota cílového atributu pro příklady v uvažovaném uzlu jen málo liší (směrodatná odchylka v tomto uzlu je menší než 5 % směrodatné odchylky pro celá data), nebo pokud je v uvažovaném uzlu jen málo příkladů (řekněme 4 a méně)[38].

[37] Rozptyl pro n příkladů se počítá podle vzorce $S_y^2 = (\sum_{i=1}^{n}(y_i - \bar{y})^2)/(n-1)$.

[38] Jedná se o ad hoc stanovené parametry, které v experimentech dávají dobré výsledky. Tyto experimenty ukazují, že výsledné stromy nejsou příliš citlivé na změnu těchto parametrů.

V uvedeném příkladu je v listech regresního stromu konstanta. Tak vypadají regresní stromy například v systému *CART* (Breiman a kol., 1984). Jiné algoritmy umožňují vyjadřovat hodnotu v listu složitěji: jako lineární kombinaci vstupů (např. systémy *RETIS* (Karalič, 1992) a *M5* (Quinlan, 1992)) nebo jako spline (systém *MARS* (Friedman, 1991)).