Klasifikace s chybějícími hodnotami atributů

Problém chybějících hodnot

V praxi může dojít k situaci, že v některých instancích nejsou vždy k dispozici hodnoty některých atributů – např. u některých pacientů nemusí být znám výsledek určitého vyšetření a u některých ano; příčinou může být ztráta dat nebo neexistence měření dané veličiny pro zkoumanou instanci, apod. Pokud daný atribut hraje roli v klasifikaci (tj. je *relevantní* z konkrétního hlediska řešené úlohy), pak příslušný test (třeba v rozhodovacím stromu nebo pravidle) neposkytne výsledek a instanci nelze zařadit do žádné kategorie. Je-li instance trénovacím příkladem, nelze ji pak k trénování použít.

Uvedený nedostatek lze řešit různými způsoby: během trénování mohou být neúplné příklady prostě vyřazeny, nebo se vyřadí u všech trénovacích instancí ty atributy, u nichž občas chybí hodnota. Zmíněné postupy mají nedostatky v tom, že buď snižují počet trénovacích příkladů (což nemusí vadit tehdy, když je dat více než dostatečné množství – praxe ale obvykle nedisponuje nadbytečným množstvím, spíše naopak) nebo jsou násilně vyřazeny některé relevantní proměnné (a tím je popis instancí neúplný). To obvykle vede k vyšší klasifikační chybě, respektive k nepoužitelnosti klasifikátoru.

Jiná řešení místo vyřazování instancí nebo atributů dávají přednost umělému doplňování chybějících hodnot. Takové doplnění může stanovit např. nejpravděpodobnější hodnotu na základě analýzy disponibilních dat v instancích (včetně zkoumání vzájemných závislostí mezi různými atributy) nebo prostřednictvím rozložení hodnot daného atributu, aj. Obecně nelze říci, která z metod je vhodnější a která ne, protože empirické testy na nejrůznějších datech ukázaly, že výsledek závisí silně na konkrétních okolnostech (množství dat, jejich typ, rozložení hodnot...) a zatímco některá metoda může často selhávat, v řadě případů zase dává dobré výsledky ve srovnání s ostatními postupy.

Příklad řešení použitého u algoritmu c4.5/c5/See5

Generátor rozhodovacích stromů c4.5 (a jeho komerční následník c5/See5, jehož demonstrační verzi lze získat na URL uvedené v [1]), který je založen na minimalizaci entropie výběrem relevantních atributů do testovacích uzlů stromu, vychází z využití určitých metod pro trénovací data a jiných pro klasifikovaná data. Přístup Australana J. Rosse Quinlana, autora c4.5/c5/See5, dává velmi dobré výsledky pro řadu nejrůznějších dat, ale samozřejmě neposkytuje absolutní záruku

bezchybného fungování ve všech případech; navíc nevylučuje alternativní řešení, která rovněž poskytují (i když také ne vždy) dobré výsledky. Zde je stručně popsán Quinlanův přístup [2] jednak proto, že je jako součást příslušného software široce využíván a dále proto, že vhodně ilustruje jeden z možných přístupů, od něhož lze v praxi očekávat většinou úspěch (s tím, že ovšem nezaručuje vždy nejlepší dosažitelný výsledek a není záruka neselhání).

Trénovací příklady

Zde je použit *pravděpodobnostní (váhový) přístup*. Vychází se z toho, že trénovací příklad $t \in T$ s danou (známou) klasifikací c_i je přiřazen do podmnožiny T_i, tj. pravděpodobnost (váha) p náležení t do třídy T_i je 1, do ostatních tříd je $p = 0$. Není-li známa hodnota některého relevantního atributu trénovacího příkladu, nelze stanovit výsledek testu této hodnoty – na výsledku testu záleží např. stanovení entropie, neboli u c4.5 zisku informace vzniklého rozdělením nehomogenní (pod)množiny na podmnožiny homogennější. Výsledky testů tedy stanovují cestu *n*-árním stromem k výslednému listu, a nelze-li v určitém uzlu stanovit exaktně kudy dál, pak jsou do nějaké míry možné všechny další cesty z daného uzlu. Jinak řečeno, vektor hodnot atributů s některými hodnotami neznámými může ukazovat do více tříd (místo do jediné, což je cílem klasifikace). Otázka je, do jaké míry zkoumaný případ náleží do každé z tříd přicházejících do úvahy. Jednotlivé cesty z daného testovacího uzlu do těchto tříd dostanou *váhy* odvozené z pravděpodobnosti náležení zkoumaného případu do příslušných tříd. Před stanovením výpočtu vah jednotlivých hran z testovacího uzlu si zaveďme podrobněji pojem *informační zisk*. Zde v principu jde o množství informace poskytované nějakým atributem. Při konstrukci stromu je vhodné jako testy vybrat ty atributy, které poskytují co největší zisk informace, tj. původní nehomogenní množina dat je testy rozdělena na řadu homogennějších (ideálně zcela homogenních) podmnožin zastupujících nakonec (v listech stromu) jednotlivé klasifikační třídy. C4.5 využívá míru informace založenou na teorii Shannona a Weavera z r. 1949.

Informační zisk

C4.5 zavádí pro konstrukci stromu tzv. *kritérium informačního zisku (information gain criterion)* založené na výběru testu vhodného atributu (pozn.: algoritmus c4.5 používá *heuristický* přístup). Předpokládejme, že je možný test s n výstupy (tj. z testovacího uzlu vychází n hran). Daná nehomogenní množina T obsahující vektory hodnot atributů (tj. příklady) je tedy dělena na pokud možno homogennější podmnožiny T_1, T_2, T_3, ..., T_n. Vyhodnocení kvality testu nepoužívá další dělení potenciálně vzniklých T_i, k dispozici je pouze rozložení tříd v T a vzniklé podmnožiny T_i. Je zapotřebí zjistit, který test (na který atribut a na jaké jeho hodno-

ty) v daném místě stromu vede k nejlepším výsledkům. Dále bude použito následující značení:

S ... libovolná množina příkladů,
|*S*| ... počet příkladů v *S*,
freq(C_i, *S*) ... počet příkladů z *S* náležejících do třídy C_i.

Náhodný výběr příkladu z *S* a jeho prohlášení za člena třídy C_j má pravděpodobnost

$$\frac{freq(C_j,S)}{|S|},$$

takže poskytovaná informace má hodnotu

$$-\log_2\left(\frac{freq(C_j,S)}{|S|}\right) \text{ bitů.}$$

(*Pozn.*: 1 *bit* je schopen poskytnout informaci *ano/ne*; protože se používají bity, má logaritmus základ 2.)

Ke zjištění míry očekávané informace, kterou podávají zprávy o tom, že určité příklady náležejí do určitých tříd, je nutno sečíst informační přínosy přes jednotlivé třídy vzhledem k jejich četnostem v *S* (*k* je počet tříd):

$$info(S) = -\sum_{j=1}^{k}\frac{freq(C_j,S)}{|S|}\times\log_2\left(\frac{freq(C_j,S)}{|S|}\right) \text{ bitů.}$$

Pokud se jako množina *S* použije množina trénovacích příkladů *T*, pak lze říci, že hodnota *info(T)* udává *průměrnou* míru informace potřebné k určení třídy pro příklad z *T*. Tato hodnota je právě tzv. *entropie* množiny *S* (či *T*). Entropie se při konstrukci rozhodovacího stromu heuristicky minimalizuje tak, aby bylo k zařazení nějakého případu do příslušné třídy zapotřebí co nejméně informace (a tedy aby vznikl co nejjednodušší strom).

Byla-li množina *T* vlivem výsledku testu *X* rozdělena na *n* podmnožin, očekávaný informační požadavek je dán jako váhovaný součet přes podmnožiny T_i:

$$info_X(T) = \sum_{i=1}^{n}\frac{|T_i|}{|T|}\times info(T_i).$$

Pak lze spočítat hodnotu míry informace získané rozdělením *T* pomocí testu *X*, tedy *informační zisk*:

$$gain(X) = info(T) - info_X(T).$$

Toto *ziskové kritérium* umožňuje vybrat takový test, který maximalizuje zisk (vzájemnou informaci mezi *X* a třídou).

Uvažme pro ilustraci následující příklad, který obsahuje záznamy hodnot *předpovědi počasí, teploty, vlhkosti* a *větru* s tím, že každý záznam (řádek v následující tabulce) má přidělenu hodnotu, zda se hrál či nehrál tenis (řádky složené z hodnot v prvních čtyřech sloupcích tvoří vektory hodnot atributů):

předpověď	**teplota [°F]**	**vlhkost [%]**	**vítr**	**třída**
slunečno	75	70	ano	hrálo se
slunečno	80	90	ano	nehrálo se
slunečno	85	85	ne	nehrálo se
slunečno	72	95	ne	nehrálo se
slunečno	69	70	ne	hrálo se
zataženo	72	90	ano	hrálo se
zataženo	83	78	ne	hrálo se
zataženo	64	65	ano	hrálo se
zataženo	81	75	ne	hrálo se
déšť	71	80	ano	nehrálo se
déšť	65	70	ano	nehrálo se
déšť	75	80	ne	hrálo se
déšť	68	80	ne	hrálo se
déšť	70	96	ne	hrálo se

Jsou zde dvě třídy a 14 trénovacích instancí; 9 případů patří do *hrálo se* a 5 do *nehrálo se*. Průměrná míra informace potřebná k přiřazení třídy k případu z *T* je:

$$info(T) = -\frac{9}{14}\times\log_2\frac{9}{14} - \frac{5}{14}\times\log_2\frac{5}{14} = 0.94 \text{ bitů.}$$

Použije-li se atribut *předpověď* k rozdělení T na tři podmnožiny, je výsledná míra informace pro *předpověď*:

$$\begin{aligned} info_X(T) &= \frac{5}{14}\times\left(-\frac{2}{5}\times\log_2\frac{2}{5} - \frac{3}{5}\times\log_2\frac{3}{5}\right) \\ &+ \frac{4}{14}\times\left(-\frac{4}{4}\times\log_2\frac{4}{4} - \frac{0}{4}\times\log_2\frac{0}{4}\right) \\ &+ \frac{5}{14}\times\left(-\frac{3}{5}\times\log_2\frac{3}{5} - \frac{2}{5}\times\log_2\frac{2}{5}\right) \\ &= 0.694. \end{aligned}$$

Zisk testem na *předpověď* je tedy

$$info(T) - info_X(T) = 0.940 - 0.694 = 0.246.$$

Použije-li se místo atributu *předpověď* jiný atribut, např. *vítr* (který dělí T na dvě podmnožiny), pak je míra informace:

$$\begin{aligned} info_X(T) &= \frac{6}{14}\times\left(-\frac{3}{6}\times\log_2\frac{3}{6} - \frac{3}{6}\times\log_2\frac{3}{6}\right) \\ &+ \frac{8}{14}\times\left(-\frac{6}{8}\times\log_2\frac{6}{8} - \frac{2}{8}\times\log_2\frac{2}{8}\right) \\ &= 0.892. \end{aligned}$$

Zisk testem na *vítr* je tedy

$$info(T) - info_X(T) = 0.940 - 0.892 = 0.048,$$

což je méně než zisk daný pomocí atributu *předpověď*, proto *předpověď* jako test má zde přednost (vzhledem ke snaze zkonstruovat co nejmenší rozhodovací strom) před atributem *vítr*.

Vyhodnocení testů

Z předchozího je zřejmé, že pokud některý testovací atribut A má chybějící hodnoty, nemůže poskytnout informaci o náležení do určité třídy pro ty instance, v nichž patřičné hodnoty nejsou k dispozici. Předpokládejme, že hodnoty atributu A jsou známy v určité části (frakci) F případů v T. Pro výpočet $info(T)$ a $info_X(T)$ se použijí pouze ty případy, kde jsou hodnoty A známy, neúplné vektory hodnot vzhledem k A jsou vynechány. Definice *zisku* je doplněna následovně:

$$\begin{aligned} gain(X) &= p(\text{A je známo}) \times (info(T) - info_X(T)) \\ &+ p(\text{A není známo}) \times 0 \\ &= F \times (info(T) - info_X(T)), \end{aligned}$$

kde p znamená pravděpodobnost. Jde tedy o *zisk* daný případy se známými hodnotami relevantního atributu *násobený* tím, jakou část tyto případy zabírají v trénovací množině – tj. pokud se některé případy musí vynechat, zisk je přirozeně snížen.

Rozdělení trénovací množiny

Test rozdělující trénovací množinu na podmnožiny se vybírá pomocí zisku modifikovaného frakcí F. Pokud je vybrán test X poskytující výstupy O_1, O_2, ... , O_n, pak v případě chybějících hodnot je nutno způsob rozdělování zobecnit. Nelze-li s jistotou přiřadit jedinou třídu, je pravděpodobnost náležení do třídy zeslabena. Příslušnost do tříd obecně ovlivňuje *váha* w: nechybí-li hodnota, je $w = 1$; chybí-li hodnota, pak není znám výstup testu a váha je pak dána pravděpodobností výstupu O_i v testovacím místě stromu. Každá podmnožina T_i nyní představuje soubor možných částečných (frakčních) případů, takže $|T_i|$ je nyní dáno součtem frakčních vah případů v množině T_i. Množina T je rozdělena na podmnožiny T_i, přičemž případy s chybějícími hodnotami patří do každé T_i s váhou w:

$$w \times p(O_i),$$

kde $p(O_i)$ je pravděpodobnost výstupu O_i a odhadne se jako podíl součtu vah případů v T majících výstup O_i vůči součtu vah případů v T, které mají známý výstup.

Klasifikace neznámého případu

Při klasifikaci neznámého případu se používá obdobný přístup: je-li dosaženo testovacího uzlu, z něhož nelze jednoznačně stanovit výstup kvůli chybějící hodnotě atributu, pak jsou vzaty do úvahy všechny možné výstupy testu a výsledkem není zařazení případu do jedné třídy, nýbrž matematické rozdělení všech tříd (listů), do nichž vedou cesty z daného testu (uzlu) – z tohoto uzlu samozřejmě nemusí být jen přímé cesty do listů, ale obecně také do dalších testovacích uzlů na nižších úrovních hierarchie stromu. Protože výsledkem je rozdělení příslušností do více tříd, bere se jako konečný výsledek třída s nejvyšší pravděpodobností příslušnosti s tím, že do ní daný neznámý případ patří jen částečně, avšak více než do jiných možných tříd.

Příklad řešení problému

K ilustraci řešení problému s chybějící hodnotou atributu je použita výše uvedená tabulka, v níž jsou trénovací příklady pro dvě třídy **hrálo se/nehrálo se** (tenis). Předpokládejme, že případ z šestého řádku

zataženo	72	90	ano	hrálo se

má neznámou hodnotu *předpovědi*, označenou otazníkem:

?	72	90	ano	**hrálo se ?**

Nyní je nutno vzít do úvahy zbývajících 13 případů, pro něž je *předpověď* známa; výsledkem jsou tyto četnosti:

	hrálo se	nehrálo se	CELKEM
slunečno	2	3	5
zataženo	3	0	3
déšť	3	2	5
CELKEM	8	5	**13**

Pomocí relativních četností lze tedy stanovit potřebné hodnoty informačních přínosů na základě kompletních případů:

$$info(T) = -\frac{8}{13}\times\log_2\frac{8}{13} - \frac{5}{13}\times\log_2\frac{5}{13}$$
$$= 0.961$$

$$info_X(T) = \frac{5}{13}\times\left(-\frac{2}{5}\times\log_2\frac{2}{5} - \frac{3}{5}\times\log_2\frac{3}{5}\right)$$
$$+\frac{3}{13}\times\left(-\frac{3}{3}\times\log_2\frac{3}{3} - \frac{0}{3}\times\log_2\frac{0}{3}\right)$$
$$+\frac{5}{13}\times\left(-\frac{3}{5}\times\log_2\frac{3}{5} - \frac{2}{5}\times\log_2\frac{2}{5}\right)$$
$$= 0.747$$

$$gain(X) = \frac{13}{14}\times(0.961 - 0.747)$$
$$= 0.199$$

Zisk (*gain*) 0.199 bitů je zde menší než byl ve výše uvedeném výpočtu pro všech kompletních 14 případů: 0.246 bitů. Pro 13 kompletních případů nepředstavuje rozdělení původní množiny problém, ale **uvedený případ s chybějící hodnotou *předpovědi* musí být přiřazen všem výsledkům dělení podle *předpovědi*** (tj. všem třem vzniklým podmnožinám T_i) s váhami odvozenými z četností, které jsou dány 13 kompletními případy (5 slunečných, 3 zatažené a 5 deštivých):

w(slunečno) = 5/13, *w(zataženo)* = 3/13, a *w(déšť)* = 5/13.

Pro *předpověď* = *slunečno* tedy vznikne podmnožina případů:

předpověď	**teplota [˚F]**	**vlhkost [%]**	**vítr**	**třída**	**váha**
slunečno	75	70	ano	hrálo se	**1**
slunečno	80	90	ano	nehrálo se	**1**
slunečno	85	85	ne	nehrálo se	**1**
slunečno	72	95	ne	nehrálo se	**1**
slunečno	69	70	ne	hrálo se	**1**
?	**72**	**90**	**ano**	**hrálo se**	**5/13**

Pro další hodnoty *předpovědi* (*zataženo, déšť*) lze snadno vytvořit dvě obdobné tabulky, přičemž řádek s hodnotou atributu *předpověď* = ? se vyskytne v každé tabulce s příslušnou váhou (*zataženo* → 3/13, *déšť* → 5/13).

Následující tabulka přináší ukázku výstupů (`*.out`) programu See5/c5 pro oba zmiňované případy: v pravém sloupci je výsledek pro kompletní data (na 6. řádku je hodnota *předpovědi* známa: *zataženo*), v levém sloupci je výsledek vytvořený pro chybějící hodnotu *předpovědi: ?*). Čísla jsou programem zaokrouhlena (přesnější údaje poskytuje program v `*.tree`; různé počítačové systémy se ovšem mohou v desetinných číslech mírně lišit). Součet případů mapovaných do listu dává celkový počet (např. 3.2+2+3.4+2.4+3=14, 4+2+3+2+3=14). Změna např. z počtu 3 instance na 3.4 instancí vznikla přidáním části neúplné instance do listu (5/13=0.3846153846≐0.4, 3/13=0.2307692308≐0.2), naopak jiným listům muselo být adekvátně ubráno. Tím, že neúplná instance je klasifikována více uzly (tj. dostane přiřazeny různé, vzájemně si odporující klasifikace), dojde přirozeně k chybě (viz číselný údaj za listem stromu typu *n/m*, kde *n* je počet případů mapovaných do daného listu a *m* je počet listem chybně klasifikovaných případů, pokud ovšem k chybám došlo). Zde byl klasifikátor testován na trénovacích datech, takže chyba může být na datech nepoužitých při učení generátoru stromů větší, zejména pro neúplné instance.

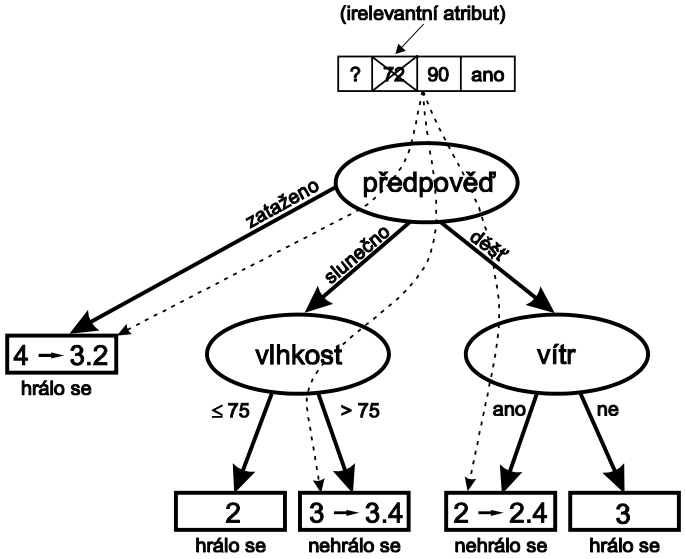
Srovnání výstupů generovaných programem See5/c5 pro uvedený příklad *tenis se hrál/nehrál*

Předpověď ? na 6. řádku

```
See5 [Release 1.14]    Sun Sep 26 14:16:23 2004
-------------------

Read 14 cases (4 attributes) from tenis.data

Decision tree:

předpověď = zataženo: hrálo se (3.2)
předpověď = slunečno:
:...vlhkost <= 75: hrálo se (2)
:   vlhkost > 75: nehrálo se (3.4/0.4)
předpověď = déšť:
:...vítr = ano: nehrálo se (2.4/0.4)
    vítr = ne: hrálo se (3)


Evaluation on training data (14 cases):

          Decision Tree
        ----------------
        Size      Errors

           5    1( 7.1%)   <<


         (a)   (b)    <-classified as
        ----  ----
           8     1    (a): class hrálo se
                 5    (b): class nehrálo se


Time: 0.0 secs
```

Předpověď *zataženo* na 6. řádku

```
See5 [Release 1.14]    Sun Sep 26 14:12:00 2004
-------------------

Read 14 cases (4 attributes) from tenis.data

Decision tree:

předpověď = zataženo: hrálo se (4)
předpověď = slunečno:
:...vlhkost <= 75: hrálo se (2)
:   vlhkost > 75: nehrálo se (3)
předpověď = déšť:
:...vítr = ano: nehrálo se (2)
    vítr = ne: hrálo se (3)


Evaluation on training data (14 cases):

          Decision Tree
        ----------------
        Size      Errors

           5    0( 0.0%)   <<


         (a)   (b)    <-classified as
        ----  ----
           9          (a): class hrálo se
                 5    (b): class nehrálo se


Time: 0.0 secs
```

Obrázek ukazuje rozhodovací strom vytvořený induktivním učením se snahou minimalizovat entropii. Proto se vůbec nedostal mezi testovací uzly druhý atribut *teplota*. Neúplná instance, znázorněná nad kořenem stromu, je přiřazena třem listům: jednou *hrálo se* a dvakrát *nehrálo se*; mapování znázorňuje čárkovaná čára. Příslušné listy ukazují, jak byl zmenšen či zvětšen počet případů (trénovacích instancí), které do nich náleží v případě úplnosti instancí. Pokud by neúplná instance byla doplněna o chybějící hodnotu předpovědi (*zataženo*), tak by bez problémů náležela do jediné třídy *hrálo se*.

Pozn. 1: Uvedený postup řešení je důvodem, proč c4.5/c5/See5 udává v případech, kdy chybí některé hodnoty atributů, počet prvků podmnožin formou necelých čísel. Podrobnější popis činnosti a výstupů See5 poskytuje program v menu *Help*; demonstrační verze programu je k dispozici ve cvičeních.

Pozn. 2: Algoritmy c4.5/c5/See5 patří k tzv. *lačným* (*greedy*) algoritmům, takže obecně nelze zaručit optimální rozhodovací strom; v praxi však jsou výsledky velmi dobré.

Reference

[1] `http://www.rulequest.com`

[2] Quinlan, J. Ross: C4.5: Programs for Machine Learning. Morgan-Kaufmann Publishers, 1993.