EVROPSKÁ UNIE
MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
OP Vzdělávání pro konkurenceschopnost
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Rozhodovací stromy a lesy

Klára Komprdová

Leden 2012

MU IBA

Příprava a vydání této publikace byly podporovány projektem ESF č. CZ.1.07/2.2.00/07.0318 „Víceoborová inovace studia Matematické biologie“ a státním rozpočtem České republiky.

Předmluva

Text byl vytvořen jako studijní materiál k přednáškám Pokročilých neparametrických metod a je určen především studentům oboru matematické biologie a dalším zájemcům z přírodovědných oborů. Nejde o „matematický text“, důraz je kladen především na pochopení principu metod, jejich interpretaci a správné použití. Teorie je doplněna konkrétními příklady v programu R. Znalost tohoto programu však není nutná pro pochopení uvedených metod.

V přírodovědných oborech se stále více začínají mimo klasických vícerozměrných parametrických technik používat i novější techniky neparametrické. Je to způsobeno především povahou biologických dat, která často nesplňují nároky na použití parametrických metod. Rovněž datové soubory nabývají na objemu a množství dat, ze kterých chceme získat zajímavou informaci, utěšeně roste. Tím však pochopitelně rostou i nároky na jejich zpracování. Text vznikl především z důvodu dosavadní nedostupnosti česky psaného materiálu ohledně těchto metod s konkrétními příklady jejich použití.

Rozhodovací stromy a lesy mohou poskytnout nový pohled na problém a doplnit zajímavou interpretaci k již stávajícím výsledkům. Je však vždy dobré si uvědomit, že metoda je pouze nástroj, který může pomoci daný problém osvětlit, ale jejíž výsledky nemůžeme brát jako absolutní pravdu.

Na tomto místě bych ráda poděkovala recenzentům, jejichž připomínky a doporučení výrazně zlepšily kvalitu těchto učebních textů.

Příprava a vydání této publikace byly podporovány projektem ESF č. CZ.1.07/2.2.00/07.0318 „Víceoborová inovace studia Matematické biologie“ a státním rozpočtem České republiky.

V Brně, leden 2012

Klára Komprdová

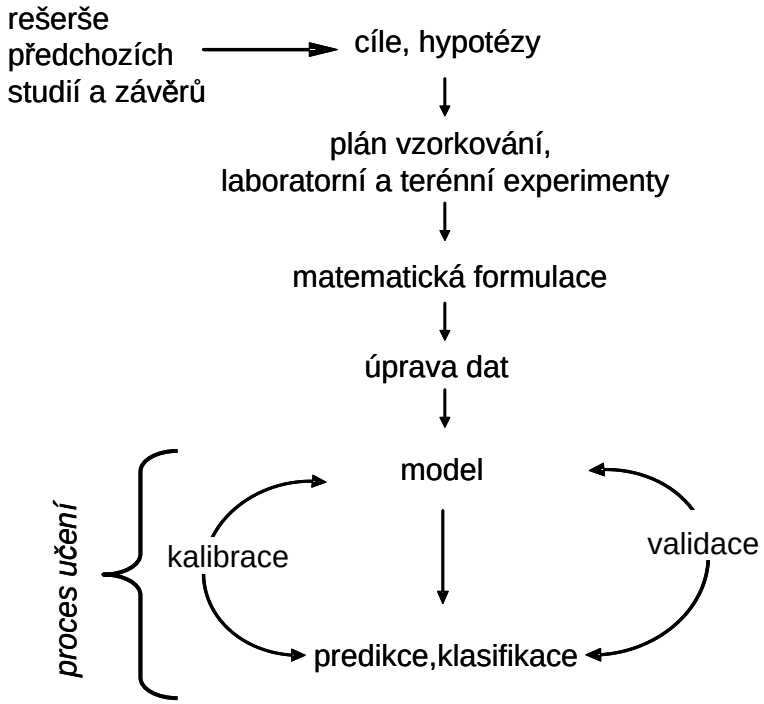
ISBN 978-80-7204-785-7

1 Úvod

Než přistoupíme ke konkrétním metodám rozhodovacích stromů a lesů, podíváme se nejdříve obecněji na proces vytváření modelu a vysvětlení základních pojmů tohoto procesu.

1.1 Proces vytváření modelu

Model popisuje vzájemné vztahy mezi pozorovanými veličinami. Na začátku je porozumění problému a nastolení otázek, na které hledáme odpověď. Následuje vytvoření plánu k jejich řešení a jeho realizace ve formě modelu a nakonec je nutná validace modelu neboli ověření jeho správnosti (obr. 1.1).



Obr.1.1 *Schéma procesu modelování.*

Z metodického hlediska je velmi důležitou částí modelování **proces učení**, který zahrnuje nastavení jednotlivých parametrů v modelu neboli **kalibraci**. Parametry se nastavují podle srovnání výsledku modelu se známými hodnotami, což je označováno jako **validace modelu**. Učení probíhá následovně: ve chvíli, kdy je model vytvořen, zjistíme jeho přesnost a obecnou platnost, následně se vrátíme v procesu o krok zpět a parametry modelu jsou změněny tak, aby se zlepšil výsledek modelu. Celý cyklus se opakuje, dokud není nalezen optimální model. Pokud již model nejsme schopni více zlepšit, proces učení končí.

Při tvorbě modelu se můžeme setkat s dalšími pojmy, které označují určitý krok v procesu modelování, zaměření modelu nebo sdělují něco o jeho vlastnostech [1][2]. Mezi základní pojmy patří:

- **simulace** - použití modelu na libovolném datovém souboru, i uměle vytvořeném. Simulace může sloužit k hlubšímu pochopení modelovaných procesů a zjištění chování modelu při limitním nastavení jeho parametrů.
- **Validace** - porovnání výsledků modelu s nezávislým datovým souborem (např. získaným experimentálně na jiné lokalitě nebo v jiném roce). Parametry modelu jsou již pevně stanoveny předchozí kalibrací. Pro pojem validace se velmi často používá také obecnější pojem testování.
- **Robustnost** - ověření funkčnosti modelu při opakované aplikaci např. za různých environmentálních podmínek a na různých lokalitách.
- **Post audit** - srovnání předpovědi výsledku modelu s experimentální činností prováděnou v budoucnosti.
- **Analýza citlivosti** - zjištění efektu malých změn parametrů modelu na jeho výsledek.
- **Analýza nejistot** - stanovení standardní odchylky predikované proměnné (jejího průměru) na základě nejistot ve vstupních parametrech modelu.
- **Expertní posouzení** - odborné zhodnocení, zda model obsahuje všechny důležité procesy a závislosti, zda jsou správně matematicky formulovány a zdali model správně popisuje modelovaný problém.
- **Tolerance k šumu** - tolerance k irelevantním neboli odlehlým pozorováním.
- **Stabilita** – model je stabilní, pokud při malé změně dat nedojde k rozdílným výsledkům modelu.
- **Predikce** – předpověď nových hodnot pomocí modelu.

Modely nemusí obsahovat všechny kroky, záleží na typu použitého modelu, možnostech datového souboru a účelu použití. Vždy by však mělo dojít k ověření výsledků modelu.

Obecné pravidlo při tvorbě modelu je vybrat co nejjednodušší model z hlediska složitosti i počtu proměnných vstupujících do modelu (princip parsimonie), který vysvětluje co největší množství informace. Výsledek je samozřejmě kompromisem mezi těmito dvěma hlavními požadavky.

1.2 Validace modelu

Validace modelu je jedním z nejdůležitějších bodů v procesu modelování a probíhá s použitím různých datových souborů.

Soubor můžeme z pohledu validace rozdělit na **trénovací** a **testovací**. Trénovací soubor je použit k tvorbě modelu a testovací soubor pro výběr nejlepšího modelu a k odhadu jeho obecné chyby. Testovací soubor by měl být nezávislý, získaný za stejných podmínek (analytických, experimentálních), ale například na jiném území (nebo jiné skupině pacientů atd.) než trénovací soubor. Ve skutečnosti většinou nenastává takto ideální situace a nezávislý testovací soubor nemusí být k dispozici. Pro tyto případy se používají různé **validační techniky**.

Máme-li k dispozici větší množství pozorování, nejjednodušší cestou je rozdělení datového souboru na dva podsoubory. Neexistuje obecné pravidlo, v jakém poměru soubor rozdělit. Časté rozdělení je 80% trénovací a 20% testovací soubor, nicméně záleží na velikosti a kvalitě datového souboru, takovéto rozdělení se nazývá jednoduché rozdělení (*simple spliting*). Většinou se však používají validační techniky, které odhadují objektivnost modelu pomocí opakovaného použití pozorování jako křížová validace (*crossvalidation*) nebo bootstrap. S jednotlivými validačními technikami se blíže seznámíme v kapitolách popisujících různé typy rozhodovacích stromů a lesů.

Validační techniky se nejčastěji používají k méně zkreslenému odhadu celkové chyby (*generalization error*) modelu a jako takové by měly být součástí všech modelů.

Odhady celkové chyby pomocí validačních technik jsou používány:

- pro výběr mezi různými modely;
- k odhadu stability modelu;
- k zjištění obecné platnosti modelu;
- k určení složitosti modelu;
- k výběru proměnných do modelu.

1.3 Typy proměnných

Proměnné se rozdělují na **kvalitativní**, **semikvantitativní** a **kvantitativní**. Hlavním kritériem je typ vztahu mezi proměnnými. U kvalitativní **(**nebo také **kategoriální**) proměnné je možné pouze určit, zda jsou dvě hodnoty stejné nebo se liší; příkladem mohou být typy půd, geologické jednotky, využití krajiny nebo textové proměnné. Dalším typem je semikvantitativní (**ordinální**) proměnná, u které můžeme určit i pořadí hodnot; sem patří abundanční skóre, řád toku nebo teplota po jednotlivých stupních. Nejširší použití má kvantitativní (také **spojitá, kontinuální**) proměnná, se kterou lze provádět prakticky všechny matematické operace. Můžeme je dále rozdělit ještě na intervalové a poměrové. Zde se dají zahrnout různé míry (výška, hmotnost), počet druhů, nadmořská výška, koncentrace atd.

Speciálním případem je proměnná **binární**, se kterou lze pracovat jako s kvantitativní, semikvantitativní i kvalitativní proměnnou. Nejčastěji se používá pro vyjádření výskytu\nevýskytu druhu.

Ze statistického hlediska dělíme proměnné na **závisle proměnnou** *Y* a **vysvětlující proměnnou** *X*, nazývanou rovněž jako **prediktor**. Závisle proměnnou se snažíme pomocí modelu vysvětlit na základě vysvětlujících proměnných. Závisle proměnnou tak může být například počet jedinců určitého druhu a vysvětlujícími proměnnými podmínky prostředí, na základě kterých se snažíme početnost odhadnout. Pro hodnoty proměnných *Y* a *X* je v textu používáno označení **pozorování**.

1.4 Rozdělení metod

Metody se dělí podle povahy zkoumaného problému na klasifikaci a regresi. U **klasifikace** se snažíme klasifikovat neznámý objekt do konečného počtu předem daných kategorií. Tímto způsobem lze např. zjistit, zda je daný druh přítomen či nepřítomen na určité lokalitě. Při tvorbě modelu se vychází z již naměřených dat. Vyberou se parametry, které jsou nejvýznamnější pro dané kategorie závisle proměnné a na základě těchto poznatků se vytvoří model, který se snaží popsat daný problém tak, aby byl schopen zařazovat neznámé vzorky.

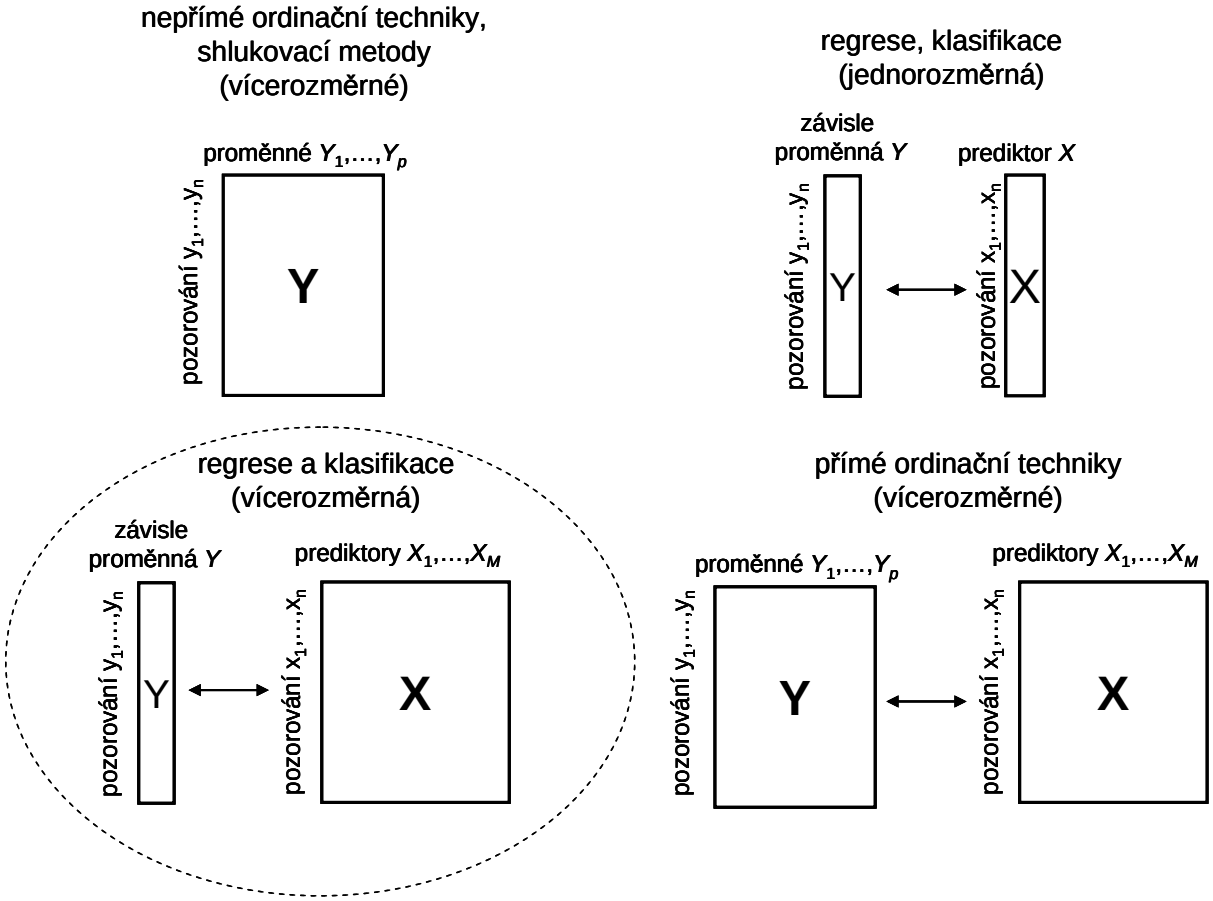
Druhou skupinou jsou modely, kdy má závisle proměnná kvantitativní charakter. V těchto případech se jedná o **regresní problém.** Při použití parametrických regresních metod získáme rovnici s odhadnutými regresními koeficienty, kterou lze použít pro předpověď hodnot závisle proměnné.

Klasifikační a regresní techniky se tedy liší především v typu závisle proměnné.

- **Klasifikační** - modelujeme závislost **kategoriální** závisle proměnné na jedné či více nezávislých proměnných.

- **Regresní** - modelujeme závislost **spojité** závisle proměnné na jedné či více nezávislých proměnných.

Další skupinou jsou modely, kdy chybí závisle proměnná, a zjišťujeme podobnost neznámého pozorování s našimi daty. Zde se často využívá především metod založených na vzdálenosti (nebo podobnosti) jednotlivých pozorování, jako jsou ordinační metody a shlukovací algoritmy (obr. 1.2).

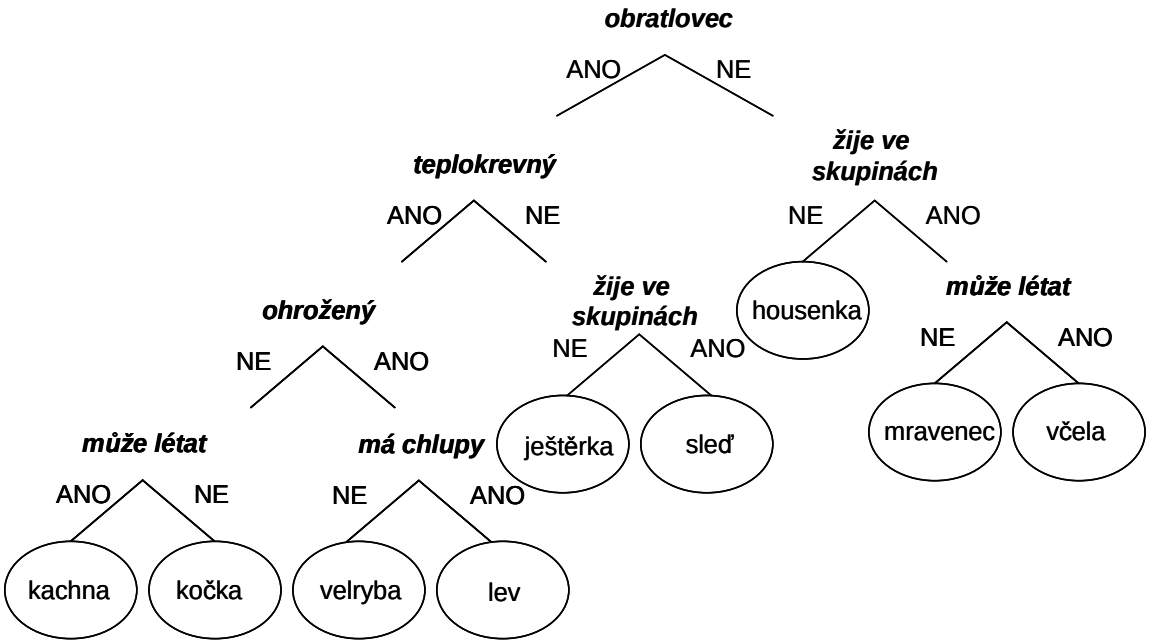


Obr. 1.2 *Rozdělení metod podle počtu závisle proměnných a prediktorů [3].*

Obsahem následujících kapitol jsou metody, které jsou podle výše uvedeného rozdělení vícerozměrnými technikami pro klasifikaci a regresi.

2 Rozhodovací stromy

Rozhodovací strom tvoří sada hierarchicky uspořádaných rozhodovacích pravidel. Příkladem jednoduchého rozhodovacího stromu může být rozdělení živočichů podle různých kritérií (obr. 2.1). Se stromovou strukturou se setkáváme poměrně často, neboť je přehledná a snadno interpretovatelná. Může jít například o rodokmeny, fylogenetické (evoluční) stromy, botanické klíče nebo zobrazení adresářů a jejich podsložek v počítači. U rozhodovacích stromů je zřejmá analogie s reálnými stromy v přírodě, proto byla jednoduše převzatá terminologie, která je pro stromy běžná a dobře vystihuje podstatu algoritmu. Podobně jako u reálného stromu tedy hovoříme o tom, že rozhodovací strom roste, větví se nebo jej prořezáváme.



Obr. 2.1 *Příklad stromu rozdělení živočichů podle kritérií zadaných v tabulce 2.1.*

Tabulka 2.1 *Kritéria pro rozdělení živočichů.*

	teplokrevný	může létat	obratlovec	ohrožený	žije ve skupinách	má chlupy
kočka	ANO	NE	ANO	NE	NE	ANO
kachna	ANO	ANO	ANO	NE	ANO	NE
sleď	NE	NE	ANO	NE	ANO	NE
lev	ANO	NE	ANO	ANO	ANO	ANO
ještěrka	NE	NE	ANO	NE	NE	NE
velryba	ANO	NE	ANO	ANO	ANO	NE
mravenec	NE	NE	NE	NE	ANO	NE
včela	NE	ANO	NE	NE	ANO	ANO
housenka	NE	NE	NE	NE	NE	ANO

Rozhodovací strom se skládá z kořene, který představuje celý soubor a postupně probíhá větvení do dalších uzlů – strom roste. Uzly, které se již dále nedělí, se označují jako terminální uzly nebo také listy. Stromy jsou binární nebo nebinární, podle toho, zda se větví na dvě nebo více větví.

Rozhodovací stromy můžeme rozdělit podle typu závisle proměnné na klasifikační a regresní.

Mějme strom T s uzly $t = (t_1,...,t_N)$. U klasifikačního stromu jsou pozorování kategoriální závisle proměnné Y s J kategoriemi zařazeny do některé z kategorií $c = (c_1,..,c_J)$, kde $J \geq 2$. Pokud je závisle proměnná spojitá $Y = (y_1,...,y_n)$, pozorováním je přiřazena hodnota predikovaná modelem $\hat{y}_i$ a výsledný strom bude regresní. Pozorování proměnné Y jsou rozdělena do uzlů hodnotami vysvětlujících proměnných (prediktorů) $X_1,...,X_M$. Pokud jsou prediktory kategoriální, jako v případě stromu na obrázku 2.1, hodnoty y_i jsou rozděleny podle kategorií prediktoru X. Například první větvení s prediktorem „obratlovec“ se dvěma kategoriemi ANO a NE rozdělí proměnnou Y do dvou dceřiných uzlů. První uzel pro hodnotu prediktoru ANO obsahuje kachnu, kočku, velrybu, ještěrku a sledě a druhý uzel pro kategorii prediktoru NE obsahuje housenku, mravence a včelu.

Rozdělení je znázorněno graficky pomocí větví stromu. Odpovídáme tedy na otázku, které pozorování y_i patří do množiny, kde $x_i \in A$, přičemž A je neprázdná vlastní podmnožina množiny všech hodnot veličiny X. V případě spojitého prediktoru rozdělujeme Y pomocí hodnoty a daného prediktoru X. V tomto případě pozorování y_i patří do prvního uzlu, pokud je $x_i \geq a$ a do druhého uzlu pokud je $x_i < a$. Takovým příkladem může být určení pohlaví dospělých koček (závisle proměnná) na základě jejich hmotnosti (prediktor). Rozdělení by mohlo dopadnout následovně: při hmotnosti $x \geq 5$ kg by se jednalo o kocoury a $x < 5$kg o kočky. Samozřejmě k přesnějšímu rozdělení těchto kategorií by bylo potřeba více prediktorů charakterizujících jednotlivá pohlaví.

K danému větvení stromu je použito vždy jen jednoho prediktoru. Stejný prediktor však může být využit v dalším větvení. Každé pozorování y_i tak patří pouze do jednoho terminálního uzlu a je mu přiřazena kategorie (klasifikační strom) nebo průměr hodnot (regresní strom) závisle proměnné Y tohoto uzlu.

Stromy nekladou nároky na rozložení dat, jako například konstantní rozptyl, normální rozložení nebo nezávislost prediktorů. Parametry algoritmu jsou často určeny experimentálně testováním různých nastavení jejich hodnot. Tento postup však skrývá nebezpečí zejména při kalibraci modelu, která může být do jisté míry subjektivní a závisí na zkušenosti badatele. Proto je při tvorbě a interpretaci modelu potřeba opatrnosti.

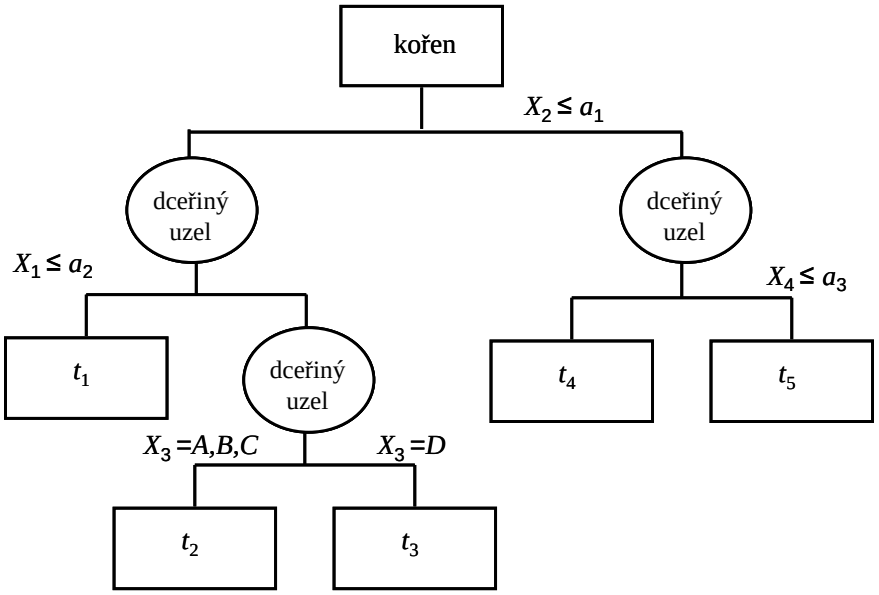
Existuje celá řada algoritmů pro vytváření stromů. Mezi jeden z nejznámějších a nejpoužívanějších patří CART, kterému se bude věnovat podkapitola 2.1 a je základním představitelem binárních stromů. Na této metodě jsou vysvětleny základní principy tvorby stromů, protože ostatní binární stromy lze získat modifikací pravidel stromu CART.

Třetí kapitola se věnuje typům stromů, které se od klasických binárních rozhodovacích stromů liší. Jde o nebinární strom typu CHAID pro kategoriální a ordinální proměnné a stromy určené pro regresní problémy PRIM a MARS. Poslední dvě metody se nedají zobrazit pomocí grafického znázornění pravidel. Výsledkem metody PRIM je sada rozhodovacích pravidel bez stromové struktury, zatímco u metody MARS je výstupem regresní rovnice.

Princip tvorby stromu je však pro všechny algoritmy velmi podobný a liší se především v nalezení vhodného prediktoru X pro každou hierarchickou úroveň stromu a hodnoty prediktoru a pro rozdělení proměnné Y. Strom na obrázku 2.1 je tedy příkladem binárního klasifikačního rozhodovacího stromu s kategoriální závisle proměnnou Y označující kategorie živočichů a kategoriálními prediktory X_i jejich vlastností.

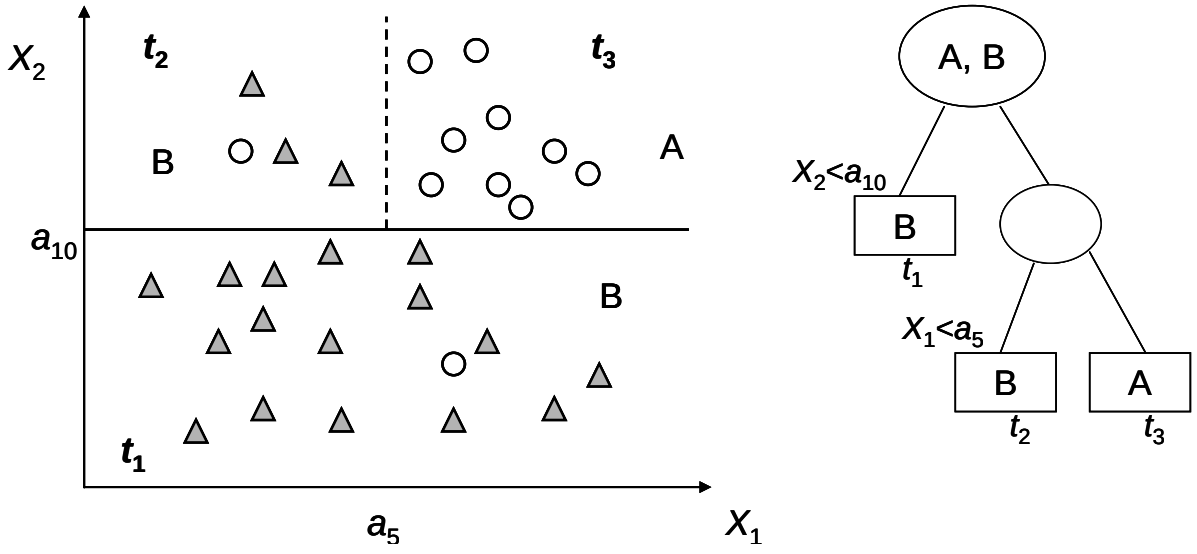
2.1 CART (*Classification and Regression Trees*)

Stromy typu CART jsou vhodné pro kategoriální i regresní úlohy a rostou na základě rekurzivního binárního dělení [4]. Na začátku tvorby stromu patří všechna pozorování souboru do jednoho uzlu neboli kořene (obr. 2.2). Následně jsou tato pozorování rozdělena do dvou dceřiných uzlů, na základě hodnoty *a* prediktoru *X*, které jsou dále děleny opět binárně na další uzly.



Obr. 2.2 *Grafická struktura rozhodovacího stromu CART. Indexy u terminálních uzlů udávají, v jakém pořadí došlo k oddělení jednotlivých terminálních uzlů. Prediktory X_1, X_2 a X_4 jsou spojité, prediktor X_4 je kategoriální s kategoriemi A,B,C,D.*

Hodnoty vysvětlujících proměnných, použité při větvení, rozdělují daný prostor na sadu pravoúhelníků (obr. 2.3), které se označují také jako regiony R_t.



Obr. 2.3 *Grafické znázornění rozdělení pozorování do kategorií A a B závisle proměnné Y s použitím dvou spojitých prediktorů X_1, X_2.*

Jak ale najít správné rozdělení? V podstatě se snažíme o takové rozdělení závisle proměnné *Y* prediktorem *X*, aby hodnoty proměnné *Y* byly uvnitř uzlu co nejhomogennější a zároveň mezi uzly co nejrozdílnější. Který prediktor (a jeho hodnota) nám zajistí nejlepší rozdělení, zjistíme pomocí tzv. **kriteriální statistiky** (*spliting criterium*), která určuje homogenitu uzlu.

2.1.1 Kriteriální statistika

Existuje několik měření kriteriálních statistik, které se navíc liší podle toho, zda se jedná o klasifikační nebo regresní strom. V následujícím textu se blíže seznámíme s nejčastěji používanými měřeními pro stromy typu CART.

Kriteriální statistika pro regresní stromy

Jak již bylo zmíněno, kriteriální statistika měří homogenitu v uzlech (*node impurity*). Předpokládejme, že máme regresní strom rozdělený do určitého počtu terminálních uzlů a predikovanou hodnotu závisle spojité proměnné *Y* chceme vyjádřit jako konstantu pro každý dceřiný uzel. Použijeme-li kritérium, které minimalizuje střední kvadratickou chybu, nejlepším odhadem této konstanty bude průměr. Snažíme se tedy najít takové rozdělení závisle proměnné *Y*, které bude mít nejmenší průměrnou kvadratickou odchylku hodnot y_i v potenciálním uzlu *t* od průměru těchto hodnot.

Kritérium minima kvadratické chyby (*Least Square Deviation*) $Q(T)$:

$$y_t = \frac{1}{N_t} \sum y_{i(t)} \tag{2.1}$$

$$Q_t(T) = \frac{1}{N_t} \sum_{i=1}^{N_t} (y_i - y_t)^2 , \tag{2.2}$$

kde N_t je počet pozorování v uzlu *t* a $y_{i(t)}$ jsou hodnoty závisle proměnné v uzlu *t*.

Kriteriální statistika pro klasifikační stromy:

Kriteriální statistika pro klasifikační stromy je založena na poměru kategorií závisle proměnné v potenciálních uzlech. Nejpoužívanějšími kritérii jsou Gini index (*GI*), Entropie (*H*) a klasifikační chyba (*ME*).

$$\text{Gini index: } GI = \sum_{c=1}^{J} p_{tc}(1 - p_{tc}) = 1 - \sum_{c=1}^{J} p_{tc}^2 \tag{2.3}$$

$$\text{Entropie: } H = -\sum_{c=1}^{J} p_{tc} \log_2 p_{tc} \tag{2.4}$$

$$\text{Klasifikační chyba: } ME = 1 - \max\{p_{tc}\} \tag{2.5}$$

kde p_{tc} je podíl pozorování y_i s kategorií *c* v uzlu *t* z celkového počtu všech pozorování y_i v tomto uzlu neboli pravděpodobnost kategorie *c* v uzlu *t*.

Gini index (*GI*) je nejčastěji používaná kriteriální statistika pro klasifikační stromy typu CART. Hodnota Giny indexu se rovná nule, pokud je v konečném uzlu pouze jediná kategorie

proměnné *Y* a dosahuje maxima, pokud je v konečném uzlu v každé kategorii proměnné *Y* stejný počet pozorování.

Ve chvíli, kdy dojde k rozdělení uzlu na dva dceřiné uzly, je *GI* spočítán pro každý dceřiný uzel. Celková hodnota Gini indexu pro dané rozdělení GI_{celk} je rovna váženému součtu všech *GI* indexů jednotlivých dceřiných uzlů. Vážení probíhá podle velikosti dceřiného uzlu. GI_{celk} tedy jednoduše spočítáme jako součet *GI (i)* dceřiných uzlů, které jsou vynásobeny příslušným podílem pozorování v daném dceřiném uzlu z celkového počtu pozorování v původním mateřském uzlu.

$$GI_{celk} = \sum_{i=1}^{K} \frac{N_i}{N_t} GI(i), \tag{2.6}$$

kde *K* je počet dceřiných uzlů (v případě binárního stromu se $K = 2$), N_t je počet pozorování v mateřském uzlu *t* a N_i jsou počty v dceřiných uzlech.

Entropie (*H*) dává velmi podobné výsledky jako Gini index. Dosahuje maxima, pokud jsou jednotlivé kategorie proměnné *Y* rovnoměrně zastoupeny v uzlech a minima pokud pozorování v uzlu náležejí pouze do jediné kategorie. Entropie je spočítána pro každý dceřiný uzel. Podobně jako u *GI* můžeme vyjádřit celkovou entropii H_{celk} pro dané dělení jako vážený součet entropií $H(i)$ v dceřiných uzlech. Entropie je často používána v algoritmu C4.5.

$$H_{celk} = \sum_{i=1}^{k} \frac{N_i}{N_t} H(i) \tag{2.7}$$

Další kritérium, které lze použít pro rozdělení stromu je *GAIN* (*information gain*, informační zisk) a měří pokles v entropii.

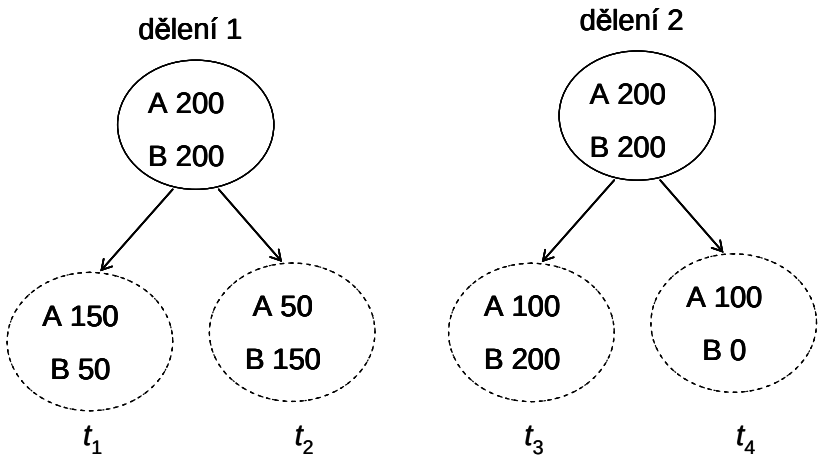
$$GAIN_{celk} = H - \left(\sum_{i=1}^{k} \frac{N_i}{N_t} H(i) \right) \tag{2.8}$$

Poslední z výše uvedených kriteriálních statistik je klasifikační chyba (*ME*). *ME* je podíl chybně klasifikovaných pozorování, čili 1 - *ME* je celková přesnost stromu (podíl správně klasifikovaných pozorování). Klasifikační chyba je obvykle používána právě k finálnímu měření přesnosti klasifikačního stromu, proto je logické její použití jako kriteriální statistiky. Na příkladu si však ukážeme, proč jsou preferovány jiné indexy. Celková klasifikační chyba pro dané dělení je opět váženým součtem *ME* v dceřiných uzlech.

$$ME_{celk} = \sum_{i=1}^{k} \frac{N_i}{N_t} ME(i) \tag{2.9}$$

Příklad I: Rozdělení uzlů

Mějme dvě kategorie závisle proměnné A a B. Počet vzorků v uzlu, který chceme rozdělit, je ve skupině A i B stejný, $n_A = n_B = 200$. Prozkoumejme nyní pomocí klasifikačních kriteriálních statistik dvě možná rozdělení stromu (obr. 2.4). Které bude vybráno jako vhodnější?



Obr. 2.4 *Ukázka možného rozdělení stromu do dvou dceřiných uzlů.*

Spočítáme hodnotu *GI* a *ME* pro každý dceřiný uzel (t_1 - t_4) možného rozdělení a následně určíme celkové indexy GI_{celk} a ME_{celk} pro obě rozdělení *D*1 a *D*2.

Tabulka 2.2 *Ukázka výpočtu kriteriálních statistik pro rozdělení stromu na obrázku 2.4.*

	uzel	n_A	n_B	p_A	p_B	p_t	$Gini = 1- p^2_A - p^2_B$	$p_t * Gini$	
D1	t_1	150	50	3/4	1/4	1/2	$1- (3/4)^2 – (1/4)^2 = 3/8$	1/2 * 3/8	0,1875
	t_2	50	150	1/4	3/4	1/2	$1- (1/4)^2 – (3/4)^2 = 3/8$	1/2 * 3/8	0,1875
								celkový	**0,375**
D2	t_3	100	200	1/3	2/3	3/4	$1- (1/3)^2 – (2/3)^2 = 4/9$	3/4 * 4/9	0,3333
	t_4	100	0	1	0	1/4	1-1-0 = 0	1/4 * 0	0
								celkový	**0,3333**

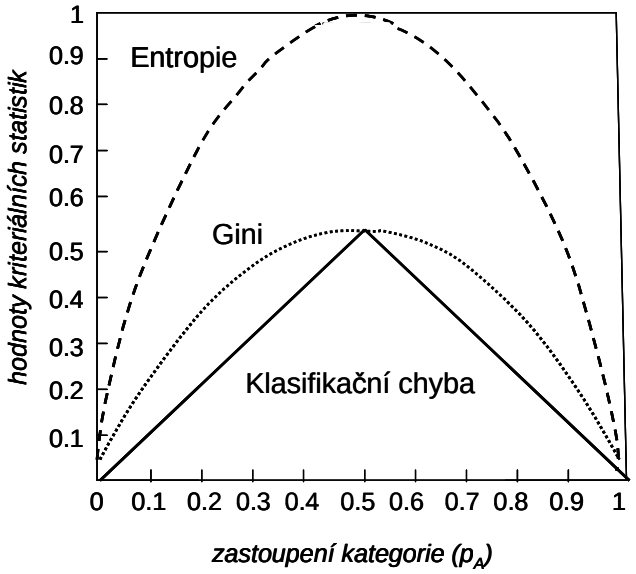
	uzel	n_A	n_B	p_A	p_B	p_t	$ME = 1- \max(p_A, p_B)$	$p_t * ME$	
D1	t_1	150	50	3/4	1/4	1/2	1- 3/4 = 1/4	1/2 * 1/4	0,125
	t_2	50	150	1/4	3/4	1/2	1- 3/4 = 1/4	1/2 * 1/4	0,125
								celkový	**0,25**
D2	t_3	100	200	1/3	2/3	3/4	1- 2/3= 1/3	3/4 * 1/3	0,25
	t_4	100	0	1	0	1/4	1-1 = 0	1/4 * 0	0
								celkový	**0,25**

V tabulce 2.2 jsou vyjádřeny počty pozorování jednotlivých kategorií v dceřiných uzlech (n_A, n_B), následně je spočítáno zastoupení kategorie v daném uzlu z celkového počtu pozorování v tomto uzlu neboli její pravděpodobnost (p_A, p_B). Poslední proměnnou, kterou budeme potřebovat při výpočtu, je podíl pozorování v dceřiném uzlu z celkového počtu pozorování

v mateřském uzlu (p_t). Hodnota Gini indexu v prvním dělení je pro oba uzly stejná (GI = 0,1875). Podobně dopadla i klasifikační chyba (ME = 0,125). Tento výsledek samozřejmě není překvapením, vzhledem k rozdělení uzlů, jejichž homogenita je stejná. Pro první rozdělení tak dostáváme celkové hodnoty indexů GI_{celk} = 0,375 a ME_{celk} = 0,25.

Druhé rozdělení $D2$ obsahuje zcela homogenní uzel t_4, ve kterém je zastoupena pouze jedna kategorie. Toto rozdělení se tedy jeví jako výhodnější. Celková hodnota Gini indexu je pro $D2$ nižší (GI_{celk} = 0,333) než pro rozdělení $D1$ (GI_{celk} = 0,375). Podle klasifikační chyby ME jsou však obě rozdělení stejně vhodné (ME_{celk} = 0,25). Na základě hodnot GI by bylo v našem příkladu vybráno rozdělení $D2$.

Důvodem, proč ME nedokázala rozlišit mezi rozděleními, je citlivost na změny v pravděpodobnostech uzlů. GI a Entropie jsou totiž mnohem více citlivé na změny v pravděpodobnostech kategorií v uzlech než ME, a proto jsou jako kriteriální statistiky vhodnější. Na obrázku 2.5 je zobrazen obecný průběh jednotlivých indexů v závislosti na pravděpodobnosti kategorie. Můžeme si všimnout, že hodnoty ME mají strmější průběh.



Obr. 2.5 *Grafické srovnání obecného průběhu kriteriálních statistik pro rozdělení do dvou kategorií A a B závisle proměnné Y jako funkce podílu první kategorie p_A. Všechny kriteriální statistiky dosahují svého maxima, pokud je kategorie rovnoměrně rozmístěna mezi uzly (p_A = 0,5) a minima, pokud je zastoupena pouze jedna kategorie ($p_A = 1$ nebo $p_A = 0 \Rightarrow p_B = 1$).*

Přiřazení hodnoty terminálnímu uzlu

U klasifikačního stromu je každému uzlu, včetně kořenového, přiřazena výsledná kategorie závisle proměnné. Výslednou kategorií je ta, která má v daném uzlu největší zastoupení. Nové pozorování je pak klasifikováno podle kategorie uzlu, do kterého je stromem zařazeno. Může se stát, že po rozdělení do dvou terminálních uzlů bude oběma uzlům přiřazena stejná kategorie, zejména je-li podíl kategorií proměnné Y nevyrovnaný. Výhodu tedy mohou mít kategorie, které jsou u proměnné Y více zastoupeny, neboť je větší pravděpodobnost, že budou mít po rozdělení v uzlu větší počet hodnot než méně početná kategorie. V takovém případě je možné použít vážení jednotlivých kategorií.

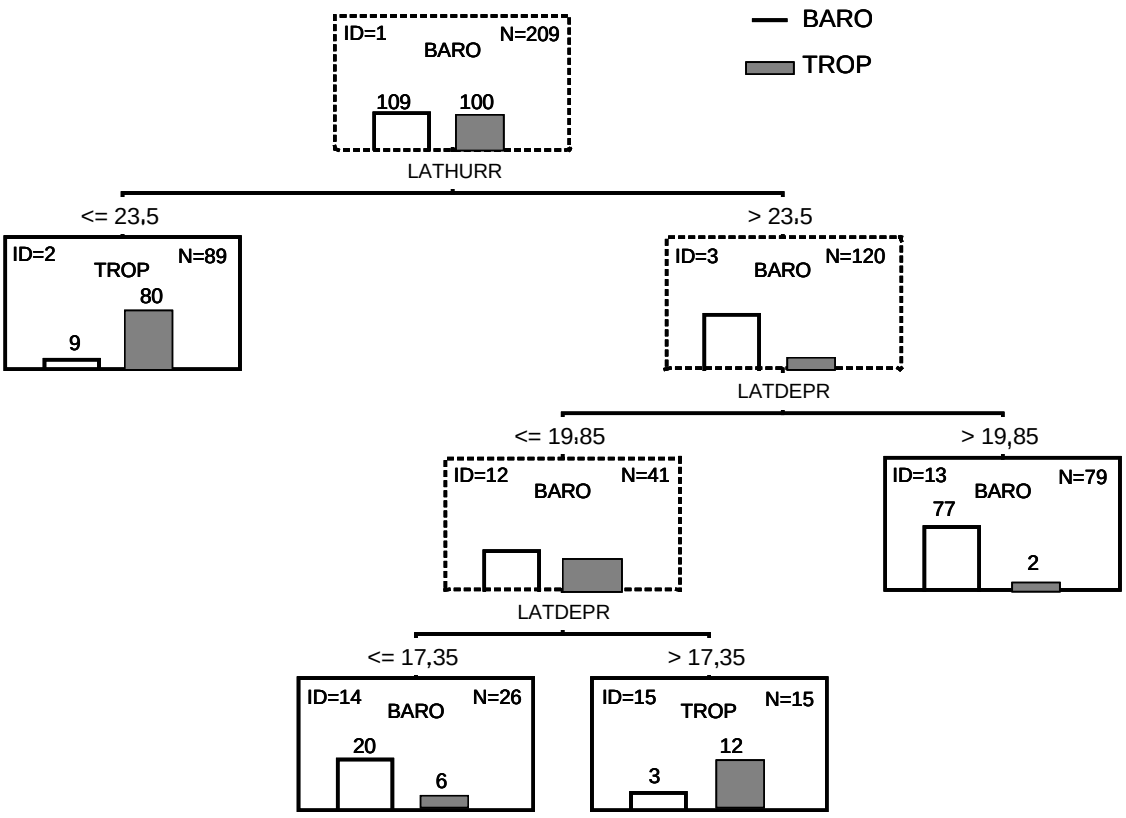
U regresních stromů je finálnímu uzlu přiřazen průměr hodnot závisle proměnné a jejich variabilita. Z variability si můžeme udělat představu, jak „přesná“ je výsledná predikovaná hodnota, jinými slovy, v jakém intervalu se může pohybovat. Při predikci stromem je novému vzorku přiřazena hodnota průměru uzlu, do kterého je zařazen. Terminální uzly lze otestovat parametrickým či neparametrickým testem, zda je mezi nimi skutečný rozdíl. Testy má však smysl provádět pouze mezi dceřinými uzly, které vycházejí ze stejného mateřského uzlu. Pokud dojde k podobným výsledným hodnotám za různých podmínek (v jiném větvení), není důvod uzly odstraňovat, neboť bychom mohli přijít o informaci a snížit přesnost stromu.

Přiřazením pouhého průměru finálním uzlům přicházíme při predikci o původní rozsah hodnot závisle proměnné. Počet výsledných predikovaných hodnot bude totiž roven počtu terminálních uzlů. Výsledná nespojitá plocha je velkou nevýhodou regresních stromů. Nespojitost lze odstranit například proložením dat lineárním regresním modelem v každém terminálním uzlu. Závisle proměnnou při regresi jsou hodnoty y_i v daném uzlu v závislosti na hodnotách x_i prediktoru X, který byl použit pro rozdělení do tohoto dceřiného uzlu. Prediktor X však musí být spojitý. Pro každý terminální uzel tak dostaneme regresní rovnici, pomocí které lze predikovat rozsah hodnot v daném uzlu. Tento postup je poměrně časově náročný, terminální uzly navíc nemusejí obsahovat dostatečný počet vzorků pro regresi, popřípadě nemusí být nalezen žádný vztah a proložení nelze uskutečnit. Tuto nevýhodu regresních stromů však řeší jiné stromové metody, např. regresní lesy nebo metoda MARS, se kterými se seznámíme v dalších kapitolách. Regresní stromy se proto používají častěji jako explanatorní technika pro vysvětlení vztahů závisle proměnné a prediktorů, než pro predikci. Další významné použití je pro nalezení mezní hodnoty nazývané diskriminační hladina (*threshold value)* při rozdělení závisle proměnné.

Následující příklad je ukázkou použití klasifikačního stromu na reálných měřeních a věnuje se rozdělení atlantických hurikánů.

Příklad II: Atlantické hurikány

Atlantické hurikány jsou klasifikovány podle ovlivnění tropickými nebo baroklinickými jevy na TROP a BARO. Tropická cyklóna prochází při vývoji třemi stádii: *tropická deprese, tropická bouře* a *hurikán*. Soubor obsahuje šest prediktorů, na základě kterých se pokusíme klasifikovat dvě třídy hurikánů. Jedná se o datum, zeměpisnou šířku a délku tropické deprese (DATEDEPR, LATDEPR, LONDEPR) a datum, zeměpisnou šířku a délku, kdy bouře dosáhla statutu hurikánu (DATEHUR, LATHUR, LONHUR). Celkem 209 měření je rozděleno na 109 BARO a 100 TROP hurikánů [5]. Ke klasifikaci hurikánů BARO a TROP použijeme klasifikační strom (obr. 2.6).

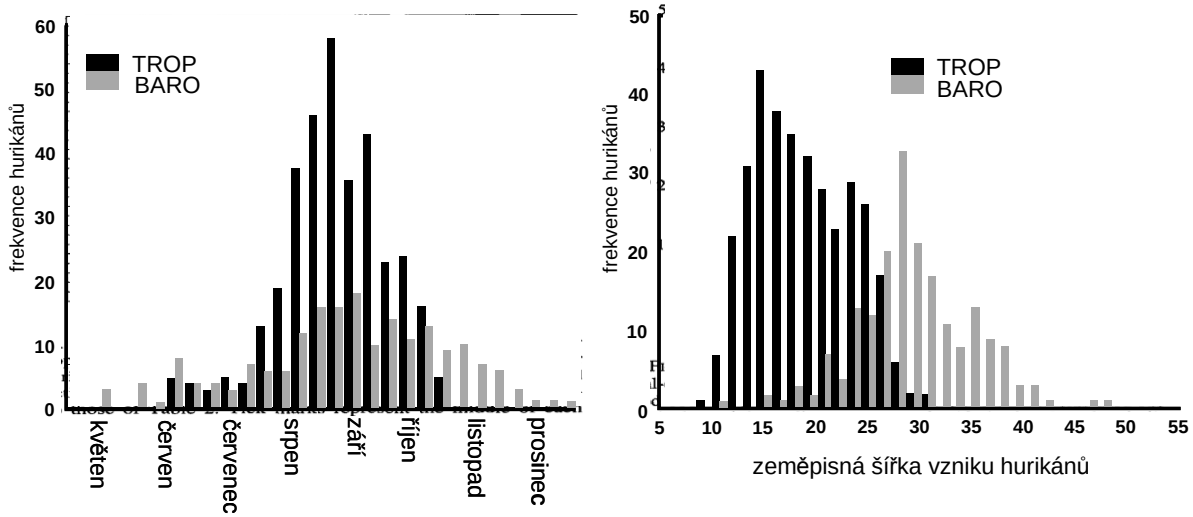


Obr. 2.6 *Klasifikační strom pro klasifikaci baroklinických a tropických hurikánů.*

Dělení stromu probíhá hierarchicky. Nejdůležitější proměnnou je zeměpisná šířka, která se objevila ve všech úrovních dělení stromu. Nejprve se hurikány rozdělí podle zeměpisné šířky, kde bouře dosáhla statutu hurikánu LATHURR. Další dělení probíhá podle zeměpisné šířky vzniku tropické deprese LATDEPR. Všimněme si, že uzlům je přiřazena převládající kategorie.

Z výsledku klasifikačního stromu vyplývá, že záleží spíše na místě vzniku hurikánu než na době jeho vzniku, protože datum se ve větvení nikde neobjevuje. Pokud místo vzniku hurikánu leží v zeměpisné šířce menší (nebo rovno) než 23,5°s.š., půjde o hurikány TROP. Pokud bouře dosáhla statutu hurikánu v zeměpisné šířce větší než 23,5° s.š. bude typ hurikánu záviset na místě vzniku jeho prvního stádia - tropické deprese. Jestliže tropická deprese vznikla mezi 17,35–19,85° s.š., je hurikán opět klasifikován jako TROP, v ostatních případech půjde o hurikán BARO. Můžeme také usuzovat, že u hurikánu TROP je méně významné, kde vznikla tropická deprese na rozdíl od hurikánů BARO a bude záviset především na zeměpisné šířce, kde se bouře změní v hurikán.

Ze srovnání histogramů hurikánů v různých měsících snadno zjistíme, že hurikány BARO se objevují v delší sezóně, od května do prosince, zatímco hurikány TROP mají nejčastější výskyt od srpna do října (obr. 2.7) [5].



Obr. 2.7 *Rozložení hurikánů BARO a TROP v období jejich výskytů a podél zeměpisné šířky vzniku hurikánů.*

Nicméně rozložení výskytu obou typů hurikánů se překrývá, a proto nebylo datum výskytu použito jako pravidlo ve stromech. Naproti tomu výskyt hurikánů podél zeměpisné šířky ukazuje jasnější rozdíl mezi hurikány. Průměrná zeměpisná šířka je pro hurikány TROP 18,8°s.š. ve srovnání s baroklinicky ovlivněnými bouřemi s průměrem 29,1° s.š..

V následujícím textu si ukážeme, jak růst stromu probíhá.

Algoritmus růstu stromu CART

1. Rozděl soubor na trénovací a testovací. Tento poměr se určuje na základě počtu pozorování a účelu studie.

2. Najdi nejlepší rozdělení každého z prediktorů:

    a. Pro spojité proměnné - seřaď hodnoty každého prediktoru (spojitého nebo ordinálního) od nejmenší po největší. Projdi všechny hodnoty prediktoru X a spočítej kriteriální statistiku všech možných rozdělení proměnné Y na dva potenciální dceřiné uzly. Pokud je dělicí hodnota a prediktoru X větší nebo rovna hodnotě x_i, pozorování y_i náleží do levého uzlu, jinak do pravého (popřípadě naopak). Hodnota a, pro kterou je kriteriální statistika minimální, je vybrána jako nejlepší možné dělení závisle proměnné Y pomocí daného prediktoru. Pro každý prediktor tak získáme jednu hodnotu (nejlepší potenciální rozdělení) kriteriální statistiky. Následně je vybrán prediktor s nejnižší hodnotou kriteriální statistiky a hodnota a je použita k rozdělení souboru (hodnot y_i) do dvou dceřiných uzlů.
    b. Pro kategoriální prediktor se za účelem nalezení nejlepšího rozdělení projdou všechny možné kombinace, tvořené jednotlivými kategoriemi prediktoru a hodnot nebo kategorií závisle proměnné. Opět se použije dělení s nejnižší hodnotou kriteriální statistiky.

3. Rozděl soubor na dva dceřiné uzly t_1 a t_2 podle hodnoty prediktoru vybrané v kroku 2.

4. Opakuj krok 2 a 3, dokud se dělení nezastaví na předem definované hodnotě (dokud není dosaženo některého z pravidel pro zastavení růstu stromu). Protože vybíráme vždy z celé množiny prediktorů, může být stejný prediktor použit ve stromě vícekrát.

5. Použij testovací soubor k ověření vhodné velikosti stromu, a pokud je strom příliš velký, prořež strom. Výběr optimálního stromu si ukážeme v následující kapitole.

Pravidla pro zastavení růstu stromu (*stopping rules*)

Strom nemůže růst donekonečna. Jeho maximální velikost je dána velikostí souboru. Jak bylo uvedeno v proceduře růstu stromu (krok 4), existují určitá pravidla, kdy se růst stromu zastaví.

Strom se zastaví sám v těchto případech:

- terminální uzel obsahuje pouze jedno pozorování;
- všechna pozorování v uzlu mají stejnou hodnotu všech prediktorů;
- všechna pozorování v uzlu mají stejnou hodnotu závisle proměnné.

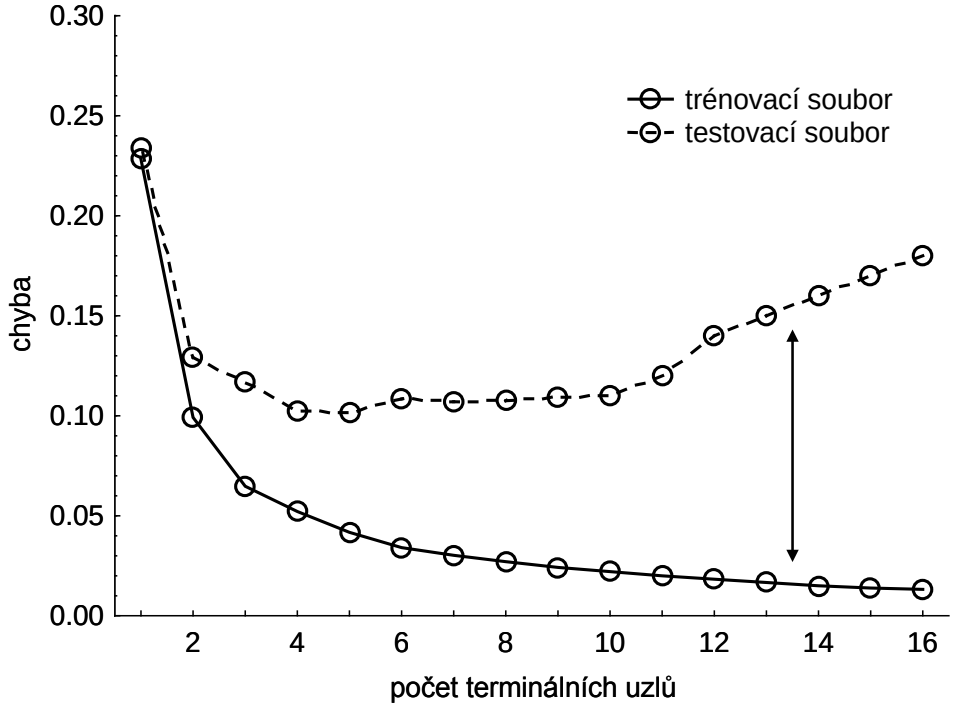
Strom můžeme v růstu omezit nastavením některých parametrů a k dalšímu rozdělení nedochází, pokud je dosaženo zadaných hodnot:

- maximální počet větvení daného stromu;
- maximální počet pozorování v koncovém uzlu;
- frakce pozorování v uzlu, která již nemůže být oddělena;
- velikosti chyby v potenciálních dceřiných uzlech - například uzel se nerozdělí, pokud střední kvadratická chyba (MSE) nebo procento nesprávně klasifikovaných vzorků v důsledku rozdělení překročí určitou hranici.

2.1.2 Výběr optimálního stromu

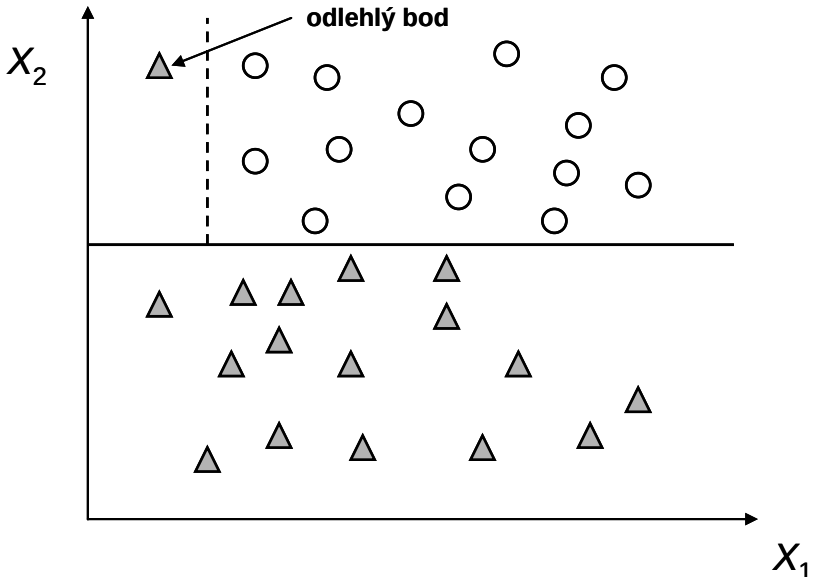
Uvedli jsme pravidla, při kterých se růst stromu zastaví. Takový strom však bude mít velikost podle námi zvolených pravidel (nebo pravidel defaultně nastavených v softwaru), která mohou být subjektivní. Jak tedy poznat, jestli je náš strom správné velikosti? Řešení spočívá v rozdělení souboru na trénovací a testovací. Na trénovacím souboru se strom učí a roste. Testovací soubor není při tvorbě stromu vůbec použit a slouží pouze k jeho otestování.

Pokud je strom nedoučený (*underfitting*), je příliš jednoduchý a chyba na testovacím i trénovací souboru bude velká. Naopak přetrénovaný (*overfitting*) strom je zbytečně složitý, trénovací chyba je většinou malá, ale testovací velká (obr. 2.8). Je tedy třeba najít vhodný kompromis.

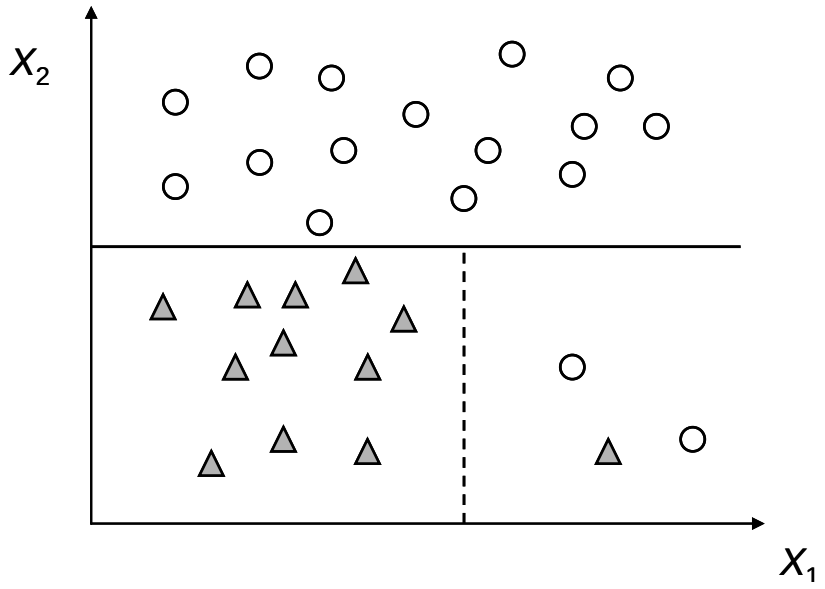


Obr. 2.8 *Rozdíl ve velikosti chyby mezi testovacím a trénovacím souborem při různé velikosti stromu, dané počtem terminálních uzlů. Nejprve byla spočítána chyba (procento chybně zaklasifikovaných pozorování) na testovacím a trénovacím souboru pro strom s 16 terminálními uzly. Postupně bylo vždy zpětně odstraněno poslední rozdělení uzlů, čímž se snížil počet terminálních uzlů o jedna. Pro takto zmenšený strom byla opět spočítána chyba pro oba soubory. Takto se postupně strom zmenšoval, až zbyl pouze jeden uzel –kořen stromu.*

Následující obrázky 2.9 a 2.10 zobrazují případy, kdy dochází k přetrénování stromu.



Obr. 2.9 *Přetrénování nastalo kvůli odlehlé hodnotě, pro kterou je vytvořeno další pravidlo a nový terminální uzel.*



Obr. 2.10 *Přetrénování z důvodu nedostatečného počtu trénovacích dat.*

Chybějící body v pravé dolní části grafu na obrázku 2.10 způsobují obtížnou predikci správné kategorie v tomto regionu. Rozhodovací strom predikuje testovací pozorování použitím trénovacích pozorování, která jsou irelevantní.

U složitějších modelů hrozí větší riziko, že dojde k přetrénování. Proto je důležité při optimální tvorbě stromu zahrnout také jeho složitost. Platí obecné pravidlo nazývané jako Occamova břitva. Pokud máme dva modely s podobnou chybou, je vhodnější vybrat ten méně složitý.

2.1.3 Prořezání stromu

Z výše uvedených příkladů je zřejmé, že parametrem, který určuje složitost stromu, je jeho velikost. Příliš velký strom ztrácí svou obecnou platnost. Na druhou stranu příliš malý strom nemusí postihnout veškerou informaci v datech. Bude tedy nutné najít optimální velikost stromu. Preferovaný přístup je nechat narůst velký strom, jako pravidlo se v tomto případě používá počet pozorování v uzlu (např. $n \leq 5$) a následně se strom prořeže.[1] K určení optimální velikosti stromu lze použít kritérium složitosti stromu (*cost-complexity criterium)*[2].

Mějme strom T_0. Prořezáním určitého počtu koncových uzlů dostaneme strom T_1.
Kritérium složitosti stromu je rovno:

$$C_\alpha(T_1) = DT_1 + \alpha|T_1|, \tag{2.10}$$

[1] U klasifikačních stromů se prořezávání nazývá *prunning*, u regresních *shrinking*, v češtině nazýváme vše obecným pojmem prořezávání.

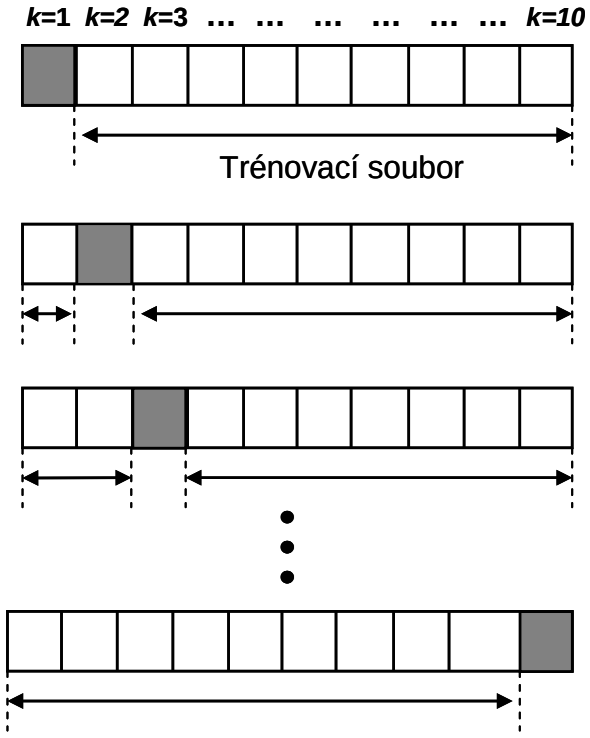
[2] Jedná se o analogii s parametrickými regresními technikami, kdy na základě AIC a BIC vybíráme nejvhodnější submodel.

kde $|T_1|$ je počet terminálních uzlů stromu a DT_1 je chyba stromu T_1. Parametr $\alpha \geq 0$ vyjadřuje kompromis mezi velikostí stromu a jeho přesností. Pro každé α hledáme takový strom $T_\alpha \subseteq T_0$, který minimalizuje $C_\alpha(T)$. K určení odhadu α se používá křížová validace (krosvalidace).

Křížová validace

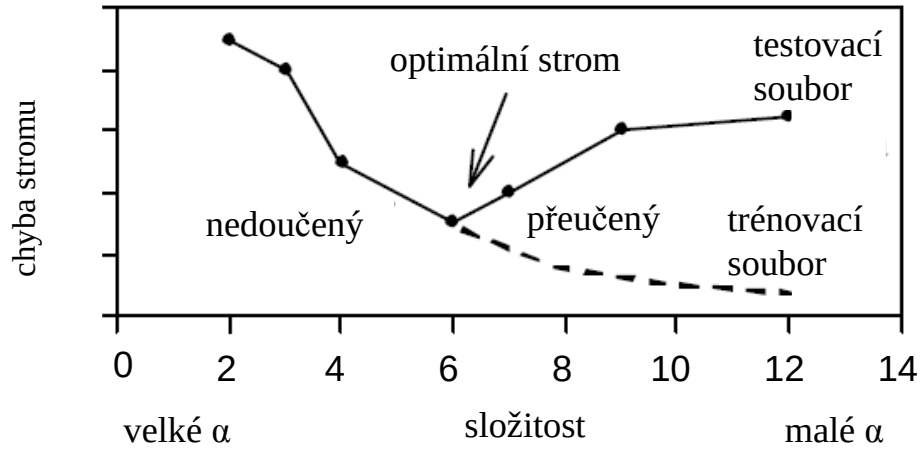
Křížová validace (*crossvalidation*) patří mezi validační techniky. Pozorování jsou rozdělena do *k* nezávislých podsouborů. Jeden podsoubor se vždy použije pro testování (pozorování nejsou použita při tvorbě modelu) všech ostatních *k*-1 skupin pro tvorbu modelu. Celkem je tedy vytvořeno *k* modelů otestovaných na *k* testovacích souborech (obr. 2.11). Z výsledků testovacích souborů můžeme určit stabilitu metody (spočítat např. průměr a směrodatnou odchylku přesnosti na testovacím souboru) a její predikční schopnost. Stromy jsou totiž obecně velmi nestabilní metody, protože i malá změna v datech může způsobit změny v rozhodovacích pravidlech a můžeme získat odlišný strom s jinou přesností. Jak velká je tato variabilita, zjistíme z rozsahu hodnot přesnosti stromu pro jednotlivé testovací soubory.

Výhoda křížové validace spočívá v použití nezávislého datového souboru pro testování (každé pozorování je pro testování použito právě jedenkrát).



Obr. 2.11 *Princip rozdělení souboru na testovací a trénovací pro k = 10. Tmavá políčka označují testovací soubor.*

Pomocí křížové validace vybereme takové α, aby měl strom co největší přesnost, ale zároveň byl rozdíl v chybě mezi testovacím a trénovacím souborem co nejmenší (obr. 2.12).



Obr. 2.12 *Hledání optimálního stromu. Složitost stromu na ose x je reprezentována počtem terminálních uzlů [6].*

2.1.4 Přesnost stromu

V předešlé kapitole jsme zjistili, že rozdělení souboru na trénovací a testovací nám může pomoci k výběru optimálního stromu. Nyní se podíváme, jak se přesnost stromu zjišťuje.

Označme *e*(*t*) chybu na trénovacím souboru (*re-substitution errors*) a *e*'(*t*) chybu na testovacím souboru (*generalization errors*). Při použití pouze trénovacího souboru lze získat dva odhady celkové chyby stromu. Optimistický odhad, kdy předpokládáme, že chyba trénovacího souboru se rovná chybě na testovacím souboru *e*'(*t*) = *e*(*t*) a pesimistický odhad, kdy je pro každý terminální uzel e'(*t*) = (e(*t*)+0,5). Celková chyba je tedy:

$$e'(T) = e(T) + N \times 0{,}5\ , \tag{2.9}$$

kde *N* je počet terminálních uzlů.

Mějme soubor obsahující 100 měření. Pro strom s 20 terminálními uzly a 10 chybně zařazenými pozorováními z trénovacího souboru je optimistický odhad chyby = 10/100 = 10% a pesimistický odhad chyb = (10 + 20×0,5)/100 = 20%.

Chyba na trénovacím souboru však není dobrým ukazatelem, jak dobře bude strom klasifikovat/predikovat nová data. Proto se k odhadu celkové (obecné) chyby stromu používá převážně testovací soubor.

Určení přesnosti klasifikačního stromu

Jedním z velmi oblíbených a nejjednodušších měření je celková správnost *OA* (*overall accuracy*), která udává pravděpodobnost, že je pozorování správně klasifikováno.

$$OA = \frac{n_p}{n}, \tag{2.11}$$

kde n_p je počet správně klasifikovaných pozorování a *n* celkový počet pozorování.

Toto měření však nezohledňuje různou velikost skupin ani rozdílnost oproti náhodnému výsledku, a proto může snadno dojít k nadhodnocení nebo naopak podhodnocení kvality modelu.

Mějme příklad klasifikačního stromu pro závisle proměnnou se dvěma kategoriemi a počtem pozorování v jednotlivých kategoriích A = 100 a B = 10. Počet správně klasifikovaných pozorování v jednotlivých kategoriích je následující A = 100 a B = 0.

$$OA = 100/110 \cong 0{,}91$$

Procento správně klasifikovaných pozorování by v tomto případě bylo zhruba 91%. Vidíme však, že takový strom nám není k užitku, protože nedokázal kategorie odlišit a všechna pozorování v kategorii B klasifikoval jako kategorii A.

Korekci na velikost kategorií lze však provést jednoduchou úpravou:

$$OA_{kateg} = \frac{1}{J}\sum_{c=1}^{J}\frac{n_{pc}}{n_c}, \qquad (2.12)$$

kde J je celkový počet kategorií, n_{pc} je počet správně klasifikovaných pozorování v kategorii c a n_c je počet všech pozorování v kategorii c.

Pro náš příklad se pak celková adjustovaná správnost stromu rovná $\frac{1}{2}\left(\frac{100}{100}+\frac{0}{10}\right)=0{,}5$.

Celková správnost se používá především pro srovnání s ostatními klasifikačními metodami nebo pro výběr vhodného stromu, v praxi nás však častěji zajímá procento správně klasifikovaných pozorování pro každou kategorii.

V příkladu II s atlantickými hurikány můžeme snadno zjistit (na základě počtu pozorování jednotlivých kategorií v terminálních uzlech), že hurikány BARO a TROP jsme schopni klasifikovat v obou případech s vysokou přesností (OA_{BARO} = 97/109 = 0,89 a OA_{TROP} = 92/100 = 0,92).

V tomto případě by se jednalo o optimistický odhad chyby $e'(t)$ = 1- OA_{tren} na trénovacím souboru. Stejný výpočet bychom však mohli provést pro pozorování z testovacího souboru a získat objektivnější měření chyby $e'(t)$ = 1- OA_{test}.

Určení přesnosti regresního stromu

U regresního stromu je přesnost, neboli variabilita vyčerpaná modelem, určována stejně jako v lineární regresi, pomocí koeficientu determinace R^2.

Koeficient determinace je obecně definován jako podíl variability závislé proměnné Y, vysvětlené modelem k celkové variabilitě proměnné Y. V našem případě jde o variabilitu vysvětlenou stromem, což je jedna mínus podíl sumy čtvercových odchylek pozorování y_i od predikované hodnoty $\hat{y}_i$ ku sumě čtvercových odchylek všech hodnot y_i od průměru $\bar{y}$ v kořenovém uzlu, který obsahuje všechna pozorování proměnné Y.

$$R^2 = \frac{variabilita\_vysvetlena\_modelem}{celkova\_variabilita\_Y} = 1 - \frac{residualni\_variabilita}{celkova\_variabilita\_Y} =$$

$$= 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}, \qquad (2.13)$$

kde $\hat{y}_i = \bar{y}_t$ je průměr v příslušných terminálních uzlech a odchylka od průměru uzlu t je spočítána vždy pro pozorování y_i zařazené do tohoto terminálního uzlu.

Koeficient determinace nabývá hodnot od 0 do 1. Při hodnotě $R^2 = 1$ jsme vysvětlili veškerou variabilitu pomocí stromu a predikované hodnoty $\hat{y}_i$ se shodují s pozorovanými hodnotami y_i.

Je opět možné spočítat chybu regresního stromu pro trénovací soubor $e(t) = 1 - R^2_{tren}$ a testovací soubor $e'(t) = 1 - R^2_{test}$.

2.1.5 Primární, zástupné a kompetitivní proměnné

Jak již bylo uvedeno, stromy jsou velmi nestabilní. Výsledný strom závisí na použité kriteriální statistice, krosvalidaci a nastavení parametrů pro zastavení růstu stromu. Pro různé trénovací soubory při krosvalidaci můžeme získat odlišné stromy. Krom těchto případů lze získat strom s jiným větvením použitím různých kombinací prediktorů. Představme si klasickou situaci, kdy je vybrán pro první dělení nejvýznamnější prediktor a strom dále pokračuje v rozdělování na další uzly. Co kdyby byl však vybrán druhý nejlepší prediktor, který by pozorování v prvním větvení rozdělil docela jinak? Pro následné rozdělení do dalších uzlů by pak byly pravděpodobně použity jiné prediktory. Tento strom však může mít stejnou přesnost jako strom původní. Výběrem vždy nejlepšího prediktoru pro rozdělení daného uzlu tak nemusíme získat strom s největší přesností[3]. Pokud však chceme zjistit, jestli je náš strom opravdu nejlepší, popřípadě najít strom podobné přesnosti, který by byl z hlediska vysvětlovaného problému lépe interpretovatelný, můžeme k tomu použít zástupné a kompetitivní proměnné.

Primární proměnná dosahuje nejlepšího dělení daného uzlu a je použita jako pravidlo ve stromě. Může se stát, že proměnná, která je téměř stejně vhodná (kriteriální statistika má podobnou hodnotu) jako vybraná primární proměnná, zůstane skrytá, i když může mít větší interpretační hodnotu. Takovéto proměnné se nazývají zástupné (*surrogates*) a kompetitivní proměnné. Zástupné proměnné nesou podobnou informaci jako primární proměnná a většinou jsou s ní korelované. Pro každý uzel lze zjistit, nakolik rozdělují pozorování v dceřiných uzlech stejně jako primární proměnná. Přesnost stromu s použitím zástupné proměnné může být v některých případech srovnatelná jako při použití primární proměnné. Zástupné proměnné mají velký význam zejména pro interpretaci.

Kompetitivní proměnná rozděluje daný uzel odlišně než primární (tab. 2.3). Na základě hodnot kriteriální statistiky se tak v případě absence primární proměnné rozdělí uzel podle kompetitivní nebo zástupné proměnné. Je tedy vybrán jiný prediktor s další nejlepší hodnotou kriteriální statistiky.

[3] Nicméně je potřeba zmínit, že ve většině případů tomu tak bývá.

Tabulka 2.3 *Určení primární, kompetitivní a zástupné proměnné při rozdělení pozorování kategorií A, B, C do dvou terminálních uzlů.*

A = 100, B = 100, C = 100

proměnná *X*	kategorie	uzel 1	uzel 2
primární	A	90	10
	B	90	10
	C	20	80
zástupná	A	80	20
	B	85	15
	C	25	75
kompetitivní	A	80	20
	B	20	80
	C	10	90

2.1.6 Výhody a nevýhody rozhodovacích stromů CART

Na závěr shrneme hlavní výhody a nevýhody rozhodovacích stromů. Velkou výhodou stromů je zejména fakt, že nekladou nároky na rozložení dat, není tedy nutné používat transformace proměnných. Jsou vhodné i pro větší počet proměnných a to všech typů. Nevýhodou je pak jejich poměrně vysoká nestabilita. Problém nestability lze odstranit například použitím lesů (kap. 5). Stromy také nedokážou postihnout aditivní vztah prediktorů, neboť je pro dělení vždy použit pouze jeden z prediktorů. Účinky jednotlivých prediktorů tak nelze „sčítat“, protože jsou hierarchicky uspořádány v různých patrech stromu. To je velký rozdíl oproti např. lineární regresi, kdy má každý prediktor stejnou váhu a jejich příspěvky sčítáme. U regresních stromů připomeňme ještě výslednou nespojitou plochu (výsledkem je průměr pro každý terminální uzel). Předchozí dvě omezení se dají odstranit např. použitím metody MARS (kap. 3.3).

Výhody

- Snadné grafické znázornění v podobě grafu se stromovou strukturou, z čehož plyne jednoduchá interpretace získaných výsledků.
- Neklade žádné podmínky na typ rozdělení závisle proměnné ani prediktorů.
- Závisle proměnná i prediktory mohou být všech typů (kategoriální, ordinální i spojité).
- Algoritmus tvorby stromu je odolný vůči odlehlým hodnotám, které lze včas odhalit při křížové validaci.
- Je možné použít korelované prediktory, protože strom roste hierarchicky a pro dělení se vybírá vždy jen jeden prediktor (mj. ze všech možných korelovaných).
- Výsledky přesnosti stromu lze snadno porovnat s výsledky jiných modelů. R^2 u regresního stromu je srovnatelný s R^2 u ostatních regresních technik a procento správně klasifikovaných pozorování s výstupy jiných klasifikačních metod.
- Je to velmi rychlá metoda při klasifikaci nových případů.
- Metoda je vhodná pro klasifikaci i regresi (pro regresi s jistými omezeními).

Nevýhody

- Nestabilita - tvar stromu velmi závisí na datech, malá změna v datech způsobí změny v rozhodovacích pravidlech uvnitř uzlů, což může vést ke změně výsledných klasifikací/predikcí.
- Vzhledem k nestabilitě je nutná opatrnost při interpretaci stromu.
- Měření přesnosti stromu je výrazně závislé na krosvalidačním mechanizmu a dalších parametrech při validaci modelu ve fázi učení (např. pravidla pro zastavení růstu stromu).
- Stromy jsou nevhodné pro malý počet vzorků a velký počet kategorií závisle proměnné.
- Vytváření stromů vyžaduje zkušenosti s nastavením parametrů v procesu validace, které je do značné míry subjektivní.

Příklad III: Regresní strom CART

Ukázkový příklad ke cvičení v programu R.[4]

V tomto příkladu budeme sledovat závislost denního měření koncentrace ozónu (ppb) na rychlosti větru (míle/h), teplotě vzduchu (denní maximum ve stupních Fahrenheita) a intenzitě slunečního záření (cal/cm^2) v New Yorku. Soubor obsahuje celkem 111 měření, která proběhla od května do září v roce 1973 [7][8].

Přízemní ozón je součástí tzv. fotochemického smogu, který se vyskytuje v místech s intenzivní automobilovou dopravou. Jeho původcem jsou oxidy dusíku emitované jako součást spalin ze spalovacích motorů. Působením slunečního záření se tyto oxidy štěpí a vzniklé radikály reagují s kyslíkem za vzniku ozónu. Jeho zvýšené koncentrace můžeme tedy očekávat v letních měsících při vyšších teplotách. Určitý nárůst koncentrací ozónu lze ale očekávat i za slunečného počasí v chladnějších měsících, pokud jsou zhoršené rozptylové podmínky. Podíváme se, zdali jsou tato očekávání ověřitelná pomocí výše zmíněných měření.

Načteme knihovnu *rpart* obsahující funkce pro stromy typu CART a knihovnu *lattice*, ve které se nachází soubor *environmental* s měřením ozónu[5]:

```
> library(lattice)
> library(rpart)
```

Načteme datový soubor a zobrazíme zdrojovou tabulku (zobrazeno pouze prvních 10 řádků):

```
> data(environmental)
> environmental
    ozone radiation temperature wind
1      41       190          67  7.4
2      36       118          72  8.0
3      12       149          74 12.6
4      18       313          62 11.5
5      23       299          65  8.6
6      19        99          59 13.8
7       8        19          61 20.1
8      16       256          69  9.7
9      11       290          66  9.2
10     14       274          68 10.9
...
```

Abychom získali lepší pohled na data a zjistili rozsahy hodnot parametrů, zobrazíme základní popisnou statistiku příkazem *summary*:

```
> summary(environmental)
     ozone          radiation      temperature         wind
 Min.   :  1.0   Min.   :  7.0   Min.   :57.00   Min.   : 2.300
 1st Qu.: 18.0   1st Qu.:113.5   1st Qu.:71.00   1st Qu.: 7.400
 Median : 31.0   Median :207.0   Median :79.00   Median : 9.700
 Mean   : 42.1   Mean   :184.8   Mean   :77.79   Mean   : 9.939
 3rd Qu.: 62.0   3rd Qu.:255.5   3rd Qu.:84.50   3rd Qu.:11.500
 Max.   :168.0   Max.   :334.0   Max.   :97.00   Max.   :20.700
```

[4] Všechny uvedené příklady k procvičení jsou součástí některé z knihovny programu R, ponechávám je proto v původním znění (např. s původními jednotkami). Totéž se týká i obrázků, tabulek a grafů, které jsou výsledkem použitých funkcí a jsou ponechány tak, jak je zobrazí R. Cílem je, aby měl čtenář možnost přímého srovnání s výsledky všech použitých příkazů.

[5] Pokud není knihovna dostupná, je potřeba ji nejdříve nainstalovat (viz příloha).

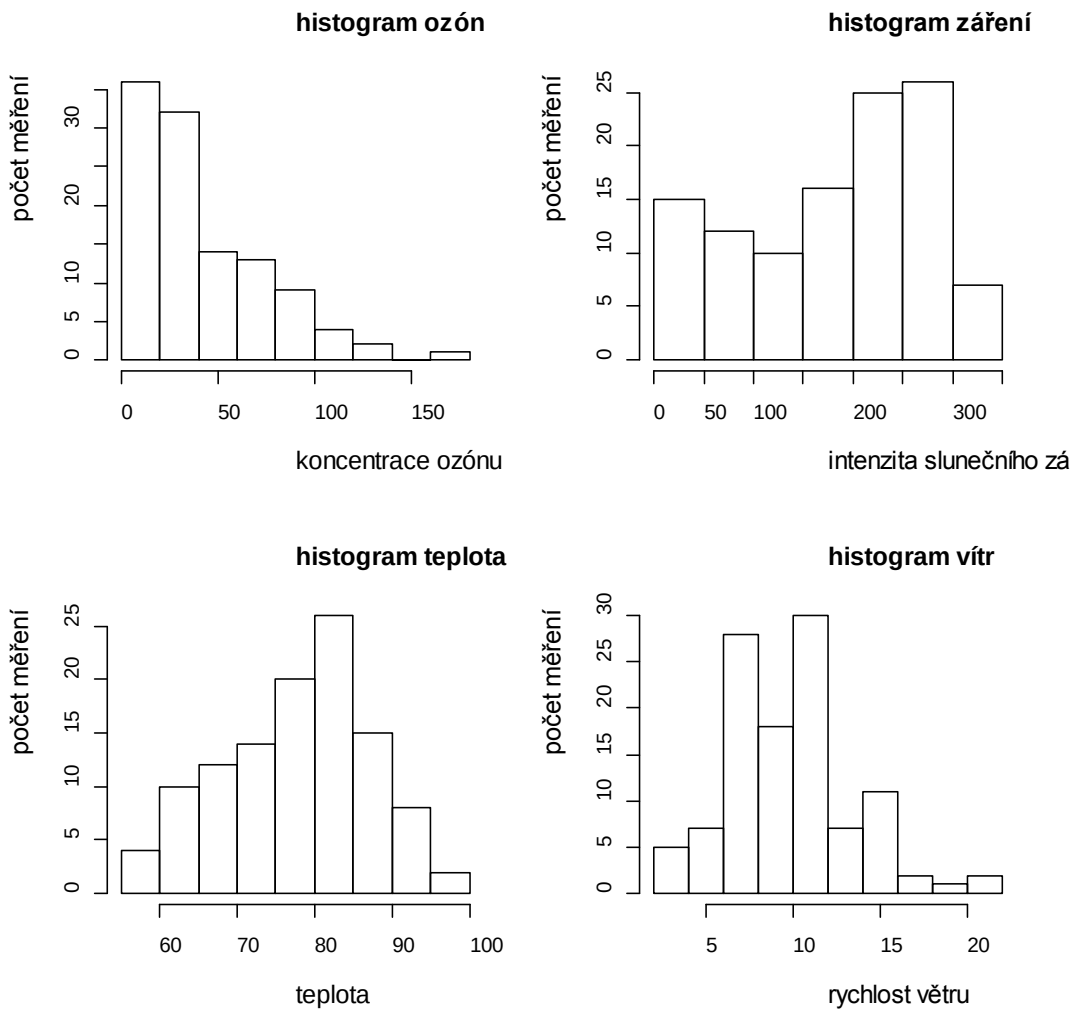
Stejným způsobem lze získat výstup pro každou proměnnou:

```
> summary(environmental$ozone)
```

Přestože regresní stromy nemají nároky na rozložení dat, podíváme se na rozložení jednotlivých proměnných. Pokud budou hodnoty odlehlé, může transformace (zejména závisle proměnné) pomoci k homogennějšímu rozdělení stromu. Pokud strom odděluje vždy pouze jeden nebo dva případy v uzlu a takto postupně „osekává“ soubor, je transformace na místě.

Rozdělíme si pole na čtyři plochy pomocí příkazu *par*, do kterých se zobrazí histogramy proměnných. Histogramy vytvoříme pomocí funkce *hist*:

```
> par(mfrow=c(2,2))
> hist(environmental $promenna)
> hist(environmental $ozone, main = paste("histogram ozón"), xlab =
   "koncentrace ozónu", ylab = "počet měření", cex.lab=1.2)
```



Z histogramů nejsou patrná odlehlá pozorování, můžeme tedy očekávat „rozumné“ rozdělení pozorování do jednotlivých uzlů. Tento příklad by bylo možné rovněž řešit (po transformaci proměnných, zejména ozónu) i klasickou lineární regresí.

Nyní vytvoříme regresní strom, který zapíšeme do proměnné *strom_ozon*. Nastavíme parametry funkce *rpart*, jako je *minsplit* - minimální počet pozorování, při kterém ještě dojde k oddělení do dalšího uzlu (pozor na hledání odlehlých hodnot) a *minbucket* – minimální počet pozorování v terminálním uzlu. Pokud jeden z těchto parametrů není zadaný, druhý z nich je

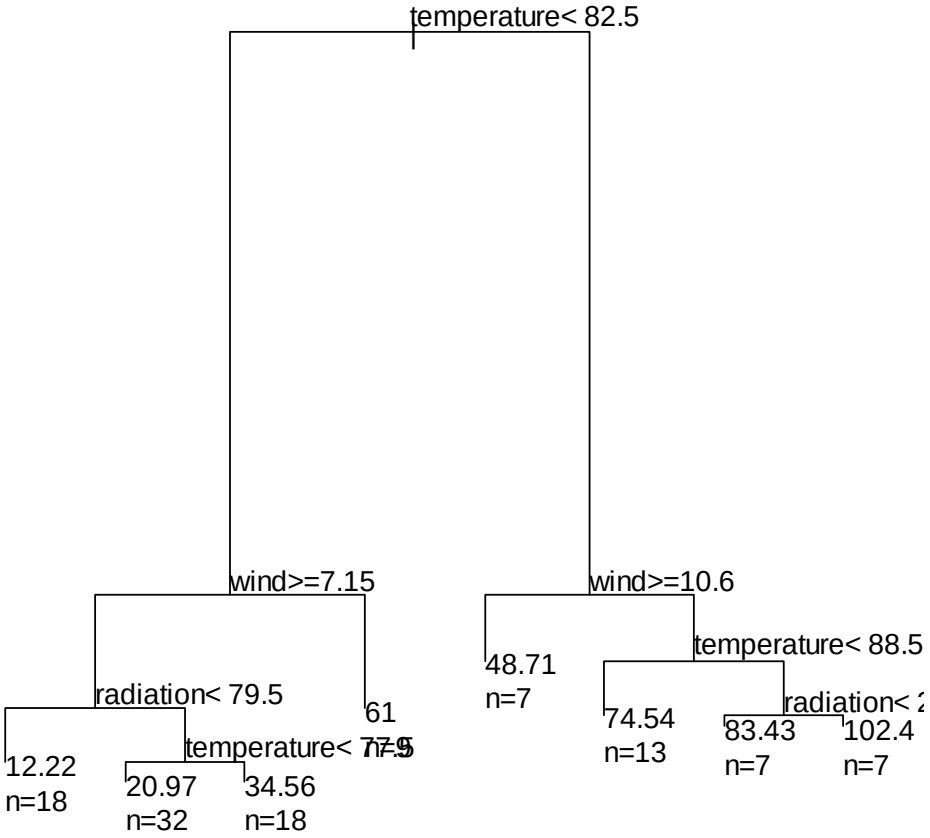
nastaven: *minbucket = minsplit/3; minsplit = 3* minbucket.* Dalším důležitým parametrem je *cp*, který určuje kritérium složitosti a parametr *xval*, což je počet podsouborů použitých při krosvalidaci neboli *k*. Ostatní parametry se vztahují k hledání počtu kompetitivních a zástupných proměnných pro každý uzel: *maxcompete* a *maxsurrogate*, defaultně jsou nastaveny na nulu.

Pomocí příkazu *plot* strom zobrazíme. Parametr *use.n = T* zobrazí počet pozorování ve výsledných uzlech, *cex* určuje velikost znaků a *margin* je parametr pro zvětšení/zmenšení okraje grafu:

```
> strom_ozon<-rpart(ozone~.,data= environmental,minsplit=10,minbucket=5,
cp =0.01)
> plot(strom_ozon,margin=0.05);text(strom_ozon, cex=1,use.n=T)
```

Vzhledem k velikosti souboru se 111-ti měřeními byl minimální počet pozorování pro rozdělení do dalších uzlů nastaven na 10, což odpovídá cca 10% z celkového souboru. Minimální počet pozorování v terminálním uzlu byl nastaven na 5. Tyto hodnoty jsou orientační, neboť optimální velikost stromu zjistíme až z krosvalidace. Nyní se snažíme pouze omezit strom tak, aby neměl maximální velikost. Rovněž parametr složitosti *cp* byl ponechán na defaultním nastavení, neboť jeho hodnotu bude nutné stanovit později z krosvalidace.

Délka větví výsledného stromu ukazuje variabilitu vyčerpanou dělením. V případě regresního stromu je to residuální suma čtverců, v případě stromu klasifikačního suma Gini indexů. Největší množství variability vysvětluje teplota.



Dělící hodnoty proměnných a příslušné operátory se vztahují vždy k levé části stromu. Je proto užitečné zobrazit si strom ještě v textové podobě, kterou obdržíme jednoduše zadáním jeho názvu:

```
> strom_ozon
n= 111

node), split, n, deviance, yval
      * denotes terminal node

 1) root 111 121801.9000  42.09910
   2) temperature< 82.5 77  42143.2500  26.77922
     4) wind>=7.15 68  10886.7500  22.25000
       8) radiation< 79.5 18    777.1111  12.22222 *
       9) radiation>=79.5 50   7648.0200  25.86000
        18) temperature< 77.5 32   2412.9690  20.96875 *
        19) temperature>=77.5 18   3108.4440  34.55556 *
     5) wind< 7.15 9  19322.0000  61.00000 *
   3) temperature>=82.5 34  20659.5600  76.79412
     6) wind>=10.6 7   1183.4290  48.71429 *
     7) wind< 10.6 27  12525.8500  84.07407
      14) temperature< 88.5 13   7307.2310  74.53846 *
      15) temperature>=88.5 14   2938.9290  92.92857
        30) radiation< 205 7    413.7143  83.42857 *
        31) radiation>=205 7   1261.7140 102.42860 *
```

Ve výsledcích je nejdříve zobrazena proměnná, která byla vybrána pro dělení, následuje hodnota rozdělení, počet pozorování v uzlu, rozptyl v daném uzlu a jeho hodnota (průměrná koncentrace ozónu). Hvězdičky označují terminální uzly, jejichž výsledné hodnoty nás zajímají nejvíce.

Všechny tři prediktory byly použity v některé z úrovní stromu. K prvnímu dělení došlo podle teploty, mezní hodnotou je 82,5°F (přibližně 28°C). Další dělení v obou větvích proběhlo podle rychlosti větru. Podívejme se nejdříve na levou větev, která obsahuje celkově nižší koncentrace ozónu (až na uzel 5 s koncentrací 61 ppb). Můžeme tedy obecně říci, že při nižších teplotách je nižší koncentrace ozónu. Nejnižší koncentrace nastává při nízké teplotě, vyšší rychlosti větru (*wind* >= 7,15 míle/h) a malé intenzitě slunečního záření (< 79,5 cal/cm^2). Pokud je však při nízké teplotě zároveň velmi nízká rychlost větru (hodnota 7,15 míle/h odpovídá zhruba 25% kvantilu- viz výstup ze *summary*), mohou průměrné koncentrace ozónu dosáhnout 61 ppb. Podíváme-li se nyní na pravou větev, k nejvyšším koncentracím dochází podle očekávání při vysoké teplotě. Lepší situace nastává jen v případě, pokud je vyšší rychlost větru. Ovšem i v tomto případě má koncentrace dvojnásobné hodnoty než při nízkých teplotách. Výsledky tedy potvrzují naše očekávání. K nejvyšší úrovni fotochemického smogu dochází ve městech s velkou dopravou, zejména v letních měsících při vysokých teplotách a na konci jara a začátkem podzimu při špatných rozptylových podmínkách. Jelikož jsou k dispozici pouze měření od května do září, nemůžeme naše výsledky porovnat se situací v zimních měsících.

Všimněme si, že v případě použití stromu pro predikci koncentrace ozónu by bylo nové pozorování zařazeno pravidly do některého z terminálních uzlů, tomuto pozorování by byla přiřazena jako výsledná hodnota průměrná koncentrace v uzlu. V případě našeho stromu mohou predikované koncentrace nabývat pouze osmi hodnot.

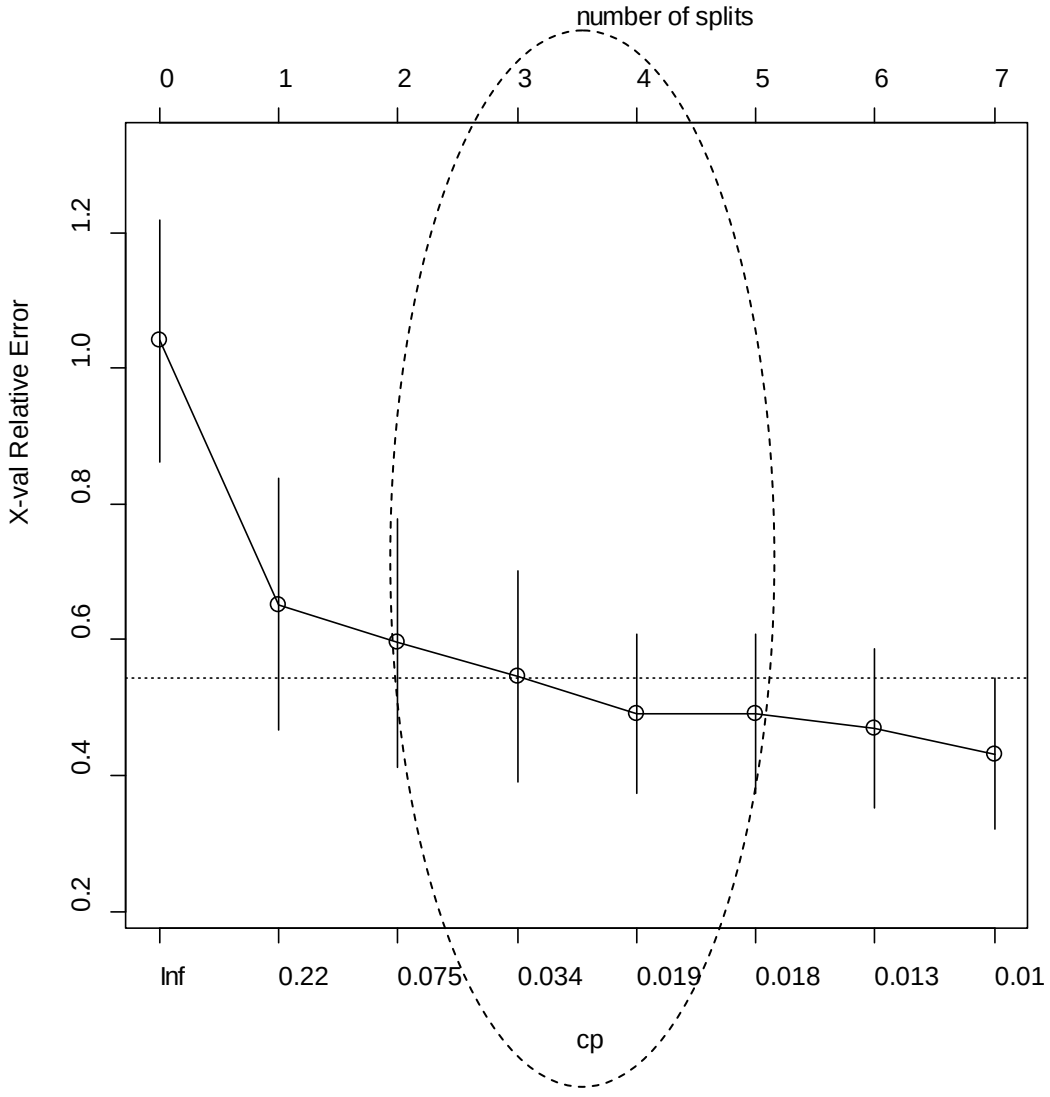
Výsledný strom bude potřeba validovat, tedy zjistit, zdali má optimální velikost. Některé terminální uzly obsahují poměrně málo pozorování a můžeme předpokládat, že strom je přetrénovaný. Otestování provedeme pomocí krosvalidace.

Krosvalidaci nastavíme ve funkci *rpart* pomocí argumentu *xval*, který určuje hodnotu *k*. Pomocí funkce *plotcp* zobrazíme závislost velikosti stromu na chybě z krosvalidace. Jde o závislost geometrických průměrů z intervalů hodnot *cp* na chybě testovacích souborů při krosvalidaci. Hodnota *cp* odpovídá α/T_1 z rovnice 2.10 a chyba na testovacím souboru je rovna $1-R^2_{test}$.

Dále můžeme specifikovat parametr *minlin*, který zobrazí (*T*) nebo nezobrazí (*F*) vodorovnou referenční čáru a parametr *upper*, který umožňuje měnit popis horní osy na počet uzlů (*size*), počet dělení (*splits*) nebo bez popisu (*none*).

Zvolíme $k = 5$, které zajistí přibližně 20 pozorování v souboru pro testování. Při nastavení např. na $k = 10$ by pro testování zbylo pouze 10 pozorování, což nemusí dostačovat pro odhad chyby. Použitím různých hodnot k lze testovat, zda rozdílné nastavení povede k výběru stromu s jinou velikostí.

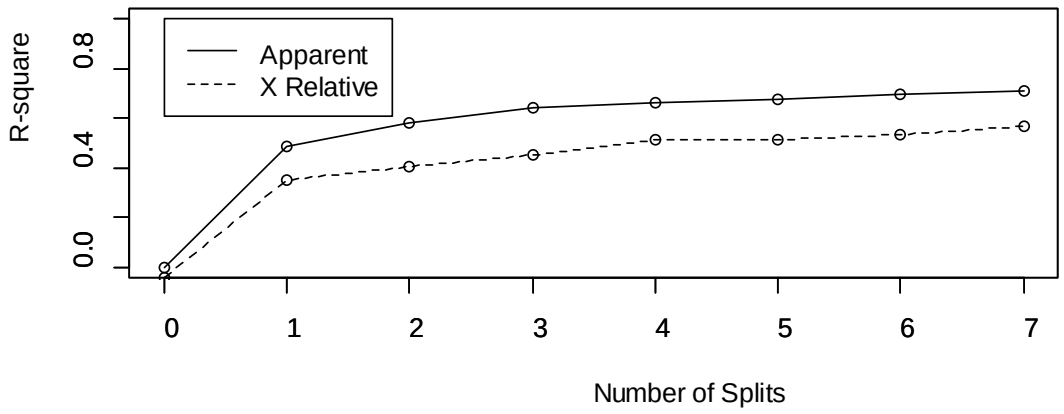
```
>  strom_ozon1<-rpart(ozone~.,data= environmental, minsplit=10,
minbucket=5, cp =0.01, xval=5)
> plotcp(strom_ozon1, minlin=T, cex.axis=1.2, cex=1.5, cex.lab=1.2, upper
=  'splits')
```



Křivka závislosti testovací chyby na *cp* postupně klesá a od velikosti stromu 3 (počet rozdělení) se již příliš nemění. Vodorovná čára je minimální krosvalidovaná chyba plus 1*SE* (standardní chyba odhadu). Doporučuje se vybrat strom, jehož průměrná hodnota *cp* leží jako první pod čarou. V našem případě 3 nebo 4.

Výsledek krosvalidace zobrazíme také v textové podobě pomocí funkce *printcp* nebo *rsq.rpart*. Použitím funkce *rsq.rpart* navíc získáme dva grafy, první zobrazuje hodnoty $e(t)$ pro trénovací a testovací soubor při různé velikosti stromu a druhý je totožný s grafem z funkce *plotcp*, ovšem bez referenční čáry.

```
> par(mfrow=c(2,1))
> rsq.rpart(strom_ozon1)

Regression tree:
rpart(formula = ozone ~ ., data = environmental, minsplit = 10,
    minbucket = 5, cp = 0.01, xval = 5)

Variables actually used in tree construction:
[1] radiation   temperature wind

Root node error: 121802/111 = 1097.3

n= 111

        CP nsplit rel error  xerror    xstd
1 0.484386      0   1.00000 1.01317 0.17319
2 0.097983      1   0.51561 0.61642 0.20061
3 0.057062      2   0.41763 0.55338 0.18230
4 0.020210      3   0.36057 0.44914 0.14573
5 0.018716      4   0.34036 0.43946 0.14518
6 0.017460      5   0.32164 0.44693 0.15807
7 0.010373      6   0.30418 0.43550 0.15827
8 0.010000      7   0.29381 0.40001 0.14004
```



V textové podobě jsou uvedeny hodnoty parametru složitosti (*cp*) a chyba na trénovacím souboru *e*'(*t*) (*rel error*). Dalším výstupem je chyba na testovacím souboru *e*(*t*) (*xerror*) a její směrodatná odchylka (*xstd*). Pozorování z testovacího souboru byla použita pouze pro validaci. Soubor byl při krosvalidaci rozdělen na pět podsouborů (*xval* = 5), z nichž jeden byl vždy použit pro tvorbu stromu (8/10 pozorování) a druhý pro jeho otestování (2/10 pozorování). Získali jsme tedy pět různých odhadů *e*(*t*), ze kterých byl spočítán průměr a odchylka.

Pro získání hodnoty R^2 musíme odečíst chybu modelu od chyby nulového modelu, který odpovídá pouze jednomu uzlu - kořenu stromu. Ve výpisu jsou to hodnoty u *nsplit* = 0. Při výběru 4. stromu je R^2_{tren} =1-0,36 = 0,64 pro trénovací soubor a R^2_{test} =1-0,45 = 0,55 pro testovací soubor. U stromu s optimální velikostí by se tyto hodnoty neměly příliš lišit.

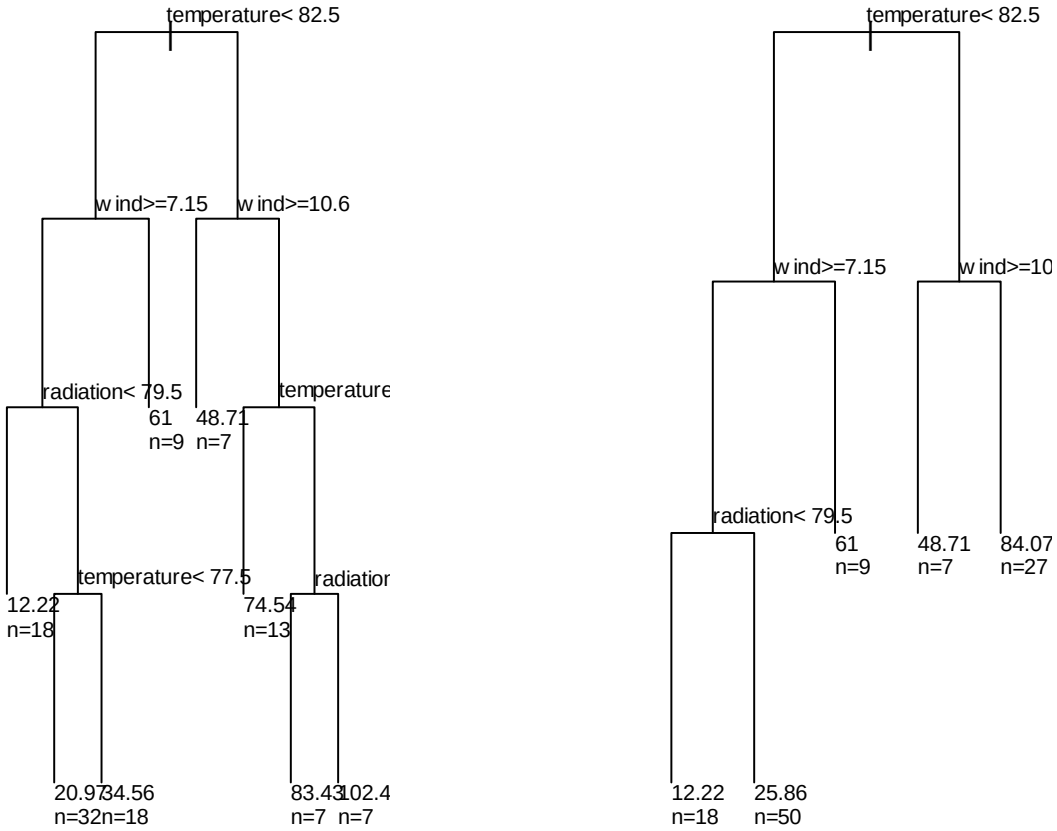
Prvně vytvořený strom s 8 terminálními uzly je přetrénovaný. Pokusíme se tedy vytvořit strom optimální velikosti. K prořezání stromu použijeme funkci *prune*. Musíme nyní specifikovat hodnotu parametru složitosti *cp*, která byla původně nastavena na velmi nízkou hodnotu 0,01. Optimální hodnotu vybereme z grafu z funkce *plotcp* nebo z textového výstupu funkce *rsq.rpart*. Hodnoty těchto parametrů se trochu liší, ale výsledný strom bude stejný. Vybereme hodnotu *cp* = 0,02, která odpovídá přibližně velikosti 4. stromu v grafu závislosti *cp*

na testovací chybě i v textovém výstupu z *rsq.rpart*(*strom_ozon1*). Výsledný strom bude obsahovat 5 terminálních uzlů:

```
> strom_ozon2<-prune(strom_ozon1,cp=0.02)
```

Nyní zobrazíme do jednoho obrázku původní a prořezaný strom. Parametr *uniform=T* zobrazí stromy se stejnou délkou jednotlivých větví:

```
> par(mfrow=c(1,2))
> plot(strom_ozon1, uniform=T,margin=0.1);text(strom_ozon1,
cex=0.64,use.n=T)
> plot(strom_ozon2, uniform=T,margin=0.1);text(strom_ozon2,
cex=0.64,use.n=T)
```



Výsledný prořezaný strom ponechal nejvýznamnější dělení podle teploty a v obou větvích podle rychlosti větru. V levé větvi s nízkými koncentracemi ozónu je také zachováno dělení podle slunečního záření, toto dělení by však v případě zvolení stromu 3 bylo odstraněno. Nyní je již na subjektivním posouzení čitatele a jeho znalosti daného problému, zda vybrat detailnější strom 4, jehož poslední dělení na základě radiace je dobře interpretovatelné, nebo si vybrat obecnější strom 3 bez tohoto rozdělení.

Zadáním příkazu *summary* s názvem vytvořeného stromu získáme, mimo chyby stromu pro testovací a trénovací soubor, také informaci o kompetitivních a zástupných proměnných:

```
> summary(strom_ozon2)

Call:
rpart(formula = ozone ~ ., data = environmental, minsplit = 7,
    minbucket = 3)
  n= 111

          CP nsplit rel error    xerror      xstd
1 0.48438571      0 1.0000000 1.0131734 0.1731877
2 0.14961639      1 0.5156143 0.6164244 0.2006107
3 0.05706215      2 0.3659979 0.4584092 0.1702192
4 0.03400000      3 0.3089358 0.3814985 0.1414337

Node number 1: 111 observations,    complexity param=0.4843857
  mean=42.0991, MSE=1097.315
  left son=2 (77 obs) right son=3 (34 obs)
  Primary splits:
      temperature < 82.5 to the left,  improve=0.4843857, (0 missing)
      wind        < 6.6  to the right, improve=0.4151041, (0 missing)
      radiation   < 153  to the left,  improve=0.2165826, (0 missing)
  Surrogate splits:
      wind < 6.6  to the right, agree=0.784, adj=0.294, (0 split)

Node number 2: 77 observations,    complexity param=0.1496164
  mean=26.77922, MSE=547.3149
  left son=4 (73 obs) right son=5 (4 obs)
  Primary splits:
      wind        < 6.6  to the right, improve=0.4324195, (0 missing)
      temperature < 77.5 to the left,  improve=0.2230477, (0 missing)
      radiation   < 153  to the left,  improve=0.1054601, (0 missing)

Node number 3: 34 observations,    complexity param=0.05706215
  mean=76.79412, MSE=607.6341
  left son=6 (7 obs) right son=7 (27 obs)
  Primary splits:
      wind        < 10.6 to the right, improve=0.3364195, (0 missing)
      temperature < 87.5 to the left,  improve=0.2895697, (0 missing)
      radiation   < 131  to the left,  improve=0.1747807, (0 missing)
  Surrogate splits:
      radiation < 93.5 to the left,  agree=0.853, adj=0.286, (0 split)

Node number 4: 73 observations
  mean=23.17808, MSE=170.2834

Node number 5: 4 observations
  mean=92.5, MSE=2872.25

Node number 6: 7 observations
  mean=48.71429, MSE=169.0612

Node number 7: 27 observations
  mean=84.07407, MSE=463.9204
```

U všech proměnných je zobrazena jejich hodnota pro rozdělení, ke kterému uzlu náleží a příspěvek tohoto rozdělení k vysvětlené variabilitě. U terminálních uzlů je uveden pouze průměr, kvadratická chyba a počet pozorování. V případě prvního uzlu by rozdělení za absence teploty proběhlo podle rychlosti větru na základě hodnoty 6,6 (míle/h). Při odstranění i proměnné rychlost větru by se uzel rozdělil podle slunečního záření. Příspěvek slunečního záření k vyčerpané variabilitě je však podstatně nižší (parametr *improve* udává procentuální změnu v součtu čtvercových odchylek y_i od průměru (SS_{uzel}) pro dané rozdělení:

1-($SS_{pravý\_uzel}$ + $SS_{levý\_uzel}$)/$SS_{mateřský\_uzel}$). Proměnné jsou tedy uvedeny v pořadí, v jakém by byly na základě kriteriální statistiky $Q_t(T)$ vybrány pro rozdělení mateřského uzlu. Výstup nerozlišuje kompetitivní proměnné, ty se však dají snadno poznat. Pokud je proměnná zároveň uvedená u *surrogate splits*, je zástupná, jinak kompetitivní. Z výstupu je zřejmé, že rychlost větru je pro první uzel zástupnou proměnnou teploty. Parametr *agree* u zástupné proměnné udává pravděpodobnost, s jakou jsou pozorování rozdělena stejně, jako u primární proměnné.

V případě druhého rozdělení (*Node number* 2) bychom další proměnnou vybírali obtížně, protože ostatní uvedené proměnné přispívají k vysvětlené variabilitě minimálně o polovinu méně než původní proměnná. Žádná z nich také není zástupnou proměnnou. Zajímavý je uzel tři. Jako další proměnná místo rychlosti větru by byla vybrána teplota, která je proměnnou kompetitivní. Až na dalším místě je sluneční záření, přestože jako zástupná proměnná odděluje 85.3% pozorování stejně jako rychlost větru. Je to dáno malým počtem pozorování v dceřiném uzlu 6 (pouze 7 pozorování). Pokud by zástupná proměnná oddělila např. 5 pozorování stejně a zbylá dvě by zařadila do druhého dceřiného uzlu, mohlo by jít o pozorování, která v tomto dceřiném uzlu významně zvýší variabilitu hodnot *Y* a tím sníží přesnost uzlu. Zástupné proměnné pro rozdělení s malým počtem pozorování v jednom nebo obou dceřiných uzlech není proto vhodné používat pro interpretaci.

3 Další metody založené na stromech – CHAID, PRIM, MARS

Jak již bylo řečeno, stromy mohou růst různými algoritmy. Dalším oblíbeným algoritmem je C5.0 a jeho dřívější verze C4.5 [9], vycházející z algoritmu ID3 [10], které jsou podobné algoritmu CART, nebo binárnímu stromu QUEST [11].

V následující kapitole si popíšeme některé algoritmy, které se od původních binárních stromů liší. Jsou jimi CHAID, PRIM a MARS.

3.1 CHAID (*Chi-squared Automatic Interaction Detector*)

Strom CHAID [12] je navržen pro kategoriální proměnné. Je často využíván v komerčních sférách, především v marketingu. Například pro optimální rozmístění zboží v obchodech podle zjištění, které zboží zákazníci nejčastěji nakupují dohromady nebo pro výběr cílové skupiny zákazníků pro různé produkty. Další oblast, kde se CHAID využívá, je hodnocení průzkumu veřejného mínění, má ale použití i v přírodovědných oborech.

CHAID je strom nebinárního typu, uzel tedy může být rozdělený na větší počet dceřiných uzlů než dva. Interpretace většího počtu uzlů je však obtížnější než u binárních stromů. Po prvním dělení také nemusí zbývat dostatek pozorování na vytvoření dalších „pater“ stromu. Tato technika je proto vhodnější pro větší datové soubory.

Kriteriální statistikou pro větvení je χ^2 –test, což vyplývá i z názvu metody. χ^2 –test je použit pro zjištění nezávislosti v kontingenční tabulce, která je tvořena kombinací kategorií závisle proměnné a prediktoru (tab. 3.1). Jsou-li Y a X nezávislé, má testová statistika přibližně Pearsonovo χ^2 rozdělení s $\upsilon = (r\text{-}1)(s\text{-}1)$ stupni volnosti, kde r je počet řádků a s je počet sloupců v kontingenční tabulce. Nezávislost v kontingenční tabulce znamená, že se obě proměnné navzájem neovlivňují v hodnotách, které nabývají. Hypotéza nezávislosti jevů je zde nulovou hypotézou H_0. Pearsonův χ^2 –test je často označován jako test dobré shody.

Tabulka 3.1 *Kontingenční tabulka.*

		kategorie prediktoru X				
	j / ***i***	**1**	**2**	**…**	***s***	**Celkem**
kategorie Y	**1**	p_{11}	p_{12}	…	p_{1s}	R_1
	2	p_{21}	p_{22}	…	p_{2s}	R_2
	…	…	…	…	…	**…**
	r	p_{r1}	p_{r2}	…	p_{rs}	R_r
	Celkem	S_1	S_2	**…**	S_s	***n***

Test dobré shody porovnává očekávané četnosti v kontingenční tabulce s jejich skutečnými četnostmi.

Výpočet χ^2 -testu je následující:

$$\chi^2 = \sum_{i=1}^{r}\sum_{j=1}^{s}\frac{\left(p_{ij} - o_{ij}\right)^2}{o_{ij}} \tag{3.1}$$

Očekávané četnosti:

$$o_{ij} = \frac{R_i S_j}{n}, \tag{3.2}$$

kde *i* a *j* je označení řádků (resp. sloupců) v kontingenční tabulce, p_{ij} je pozorovaná frekvence, o_{ij} očekávaná frekvence, *n* je celkový počet pozorování, R_i je počet pozorování v řádku *i*, S_j je počet pozorování ve sloupci *j*.

Výpočet kriteriální statistiky pro CHAID si ukážeme na následujícím příkladu.

Příklad IV

Bylo zkoumáno celkem 160 semen dvou druhů příbuzných rostlin. Semena byla roztříděna do následujících kategorií: žluté/hladké; žluté/vrásčité; zelené/hladké; zelené/vrásčité (tab. 3.2).

Tabulka 3.2 *Rozdělení semen dvou příbuzných rostlin podle barvy a tvaru.*

	žluté/hladké	**žluté/vrásčité**	**zelené/hladké**	**zelené/vrásčité**	**Celkový součet**
Druh1	10	25	10	15	60
Očekávaný počet	11,25	20,63	11,25	16,87	60
Druh2	20	30	20	30	100
Očekávaný počet	18,75	34,37	18,75	28,13	100
Celkový součet	30	55	30	45	160

$$\chi^2 = \frac{(10-11{,}25)^2}{11.25} + \frac{(25-20{,}63)^2}{20.63} + \frac{(10-11{,}25)^2}{11.25} + \frac{(15-16{,}87)^2}{16{,}87}$$

$$+\frac{(20-18{,}75)^2}{18{,}75} + \frac{(30-34{,}37)^2}{34{,}37} + \frac{(20-18{,}75)^2}{18{,}75} + \frac{(30-28{,}13)^2}{28{,}13} \approx 2.26$$

$\upsilon = (r\text{-}1)(s\text{-}1) = 3$

Abychom mohli zamítnout hypotézu H_0 o nezávislosti proměnných, porovnáme hodnotu testové statistiky s kritickou hodnotou (kvantilem) příslušné hladiny významnosti $\alpha = 0{,}05$.

V tabulce pro χ^2 rozdělení s příslušnými stupni volnosti nalezneme hodnotu:

$$\chi^2_{(1-\alpha)}{}^{(\nu)} = \chi^2_{(0{,}95)}{}^{(3)} = 7{,}81$$

Nulovou hypotézu nemůžeme zamítnout. Nelze prokázat, že barva a tvar semene jsou rozdílné mezi druhy rostlin.

Algoritmus růstu stromu CHAID

Algoritmus si předvedeme na ilustračním příkladu. Zajímá nás klasifikace potravních strategií druhů makrozoobentosu podle různých kategorií nadmořské výšky. Pro jednoduchost se budeme zabývat pouze jednou proměnnou.

Pro každý prediktor X_i:

1. Vytvoř kontingenční tabulku kategorií závisle proměnné a prediktoru (tab. 3.3).

Tabulka 3.3 *Kontingenční tabulka kategorií nadmořské výšky a potravních strategií druhů. V buňkách by byly počty jednotlivých druhů, mající vlastnosti obou kategorií.*

	N - nížinné (< 300 m)	S - střední (301 – 500 m)	P - podhorské (501 – 700 m)	H – horské (> 701 m)
sběrači				
spásači				
filtrátoři				
dravci				

2. Nyní mohou nastat tři případy:
   a. Pokud je počet kategorií prediktoru > 2, utvoří se dvojice z kategorií prediktoru. Najde se taková dvojice, která si je co do hodnot závisle proměnné *Y* nejvíce podobná neboli dvojice, jejíž χ^2 - test má nejvyšší *p* hodnotu.
   b. Pokud má prediktor 2 kategorie, algoritmus pokračuje krokem 5.
   c. Pokud má prediktor *X* pouze jednu kategorii, pak je *p* hodnota nastavena na 1.

3. Dvojice s nejvyšší *p* hodnotou, která není statisticky významná (nebo jejichž *p* hodnota je větší než α zadaná uživatelem (*alpha2*) viz příklad V), se sloučí do jedné skupiny (tab. 3.4). U ordinálních kategorií se spojují pouze sousední kategorie, u kategoriálních jsou dvojice vytvořeny kombinací všech kategorií. Prediktor *X* je dále používán s novými již sloučenými kategoriemi (tab. 3.5). Pokud je i po sloučení počet kategorií > 2, algoritmus se vrátí do kroku 2. Pokud ne, algoritmus pokračuje krokem 5.

Tabulka 3.4 *Ukázka kroku 2 a 3. Pro každou podtabulku je spočítán Pearsonův χ^2 -test nezávislosti. Najdeme největší p hodnotu testu, pokud není signifikantní (menší než zvolené α), kategorie spojíme. Protože je nadmořská výška ordinální parametr, můžeme sloučit pouze vedlejší kategorie.*

	N	S
sběrači		
spásači		
filtrátoři		
dravci		

χ_1^2 $p = 0{,}01$

	S	P
sběrači		
spásači		
filtrátoři		
dravci		

χ_1^2 $p = 0{,}05$

	P	H
sběrači		
spásači		
filtrátoři		
dravci		

χ_1^2 $p = 0{,}1$

	N	S	P + H
sběrači			
spásači			
filtrátoři			
dravci			

4. Sloučené kategorie mohou být zpětně rozděleny. Jestliže se nově vytvořené skupiny kategorií skládají ze tří nebo více původních kategorií, najde se nejlepší binární rozdělení mezi sloučenými kategoriemi (s nejnižší *p* hodnotou). Pokud je *p* hodnota významná (nebo menší než α hodnota zadaná uživatelem (*alpha3*)), dojde k rozdělení a algoritmus se vrátí do kroku 2.

5. Každá kategorie, která má velmi málo pozorování (minimum je definováno uživatelem), je spojena s nejpodobnější kategorií (opět určeno na základě největší *p* hodnoty). Toto nastavení je volitelné a bývá dostupné jen v některých softwarech.

Výše popsaným postupem jsme získali optimální sloučení pro každý prediktor (tab. 3.5).

Tabulka. 3.5 *Test sloučených kategorií. Opět spočítáme Pearsonův χ2-test nezávislosti pro každou podtabulku, nyní již sloučených kategorií. Obě p hodnoty byly statisticky významné pro zvolené α=0,05 a k dalšímu sloučení již nedochází. Přecházíme rovnou do kroku 6, neboť jsme získali optimální sloučení prediktoru (krok 4 a 5 není v našem příkladu potřeba).*

	N	S
sběrači		
spásači		
filtrátoři		
dravci		

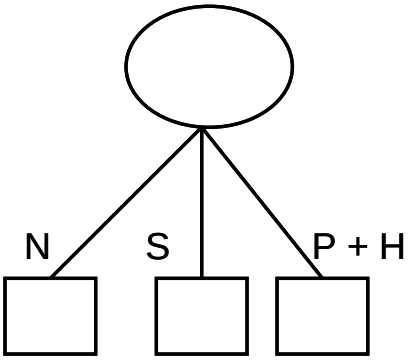
χ_1^2 p = 0,01

	S	P + H
sběrači		
spásači		
filtrátoři		
dravci		

χ_1^2 p = 0,001

	N	S	P + H
sběrači			
spásači			
filtrátoři			
dravci			

6. V posledním kroku je spočítána adjustovaná *p* hodnota χ^2 testu pro sloučené kategorie každého z prediktorů pomocí Bonferroniho korekce (viz níže). Vybere se prediktor s nejmenší adjustovanou *p* hodnotou (nebo hodnotou, která je menší/rovna hodnotě α definované uživatelem (*alpha4*)). Tento prediktor s optimálně sloučenými kategoriemi je použit k rozdělení uzlu (obr. 3.1). Pokud významný prediktor nelze nalézt, uzel se již dále nedělí a je považován za terminální.



Obr. 3.1 *Finální rozdělení uzlu. Za předpokladu, že je nadmořská výška prediktorem s nejnižší adjustovanou p hodnotou, původní uzel obsahující celý datový soubor bude rozdělen na tři dceřiné uzly podle sloučených kategorií nadmořské výšky.*

V algoritmu dochází k současnému testování více hypotéz, v našem příkladu bylo třeba učinit celkem čtyři testy pro možné sloučení kategorií. Při mnohonásobném testování však vzrůstá pravděpodobnost, že zamítneme nulovou hypotézu H_0, přestože platí. Počet prováděných testů u metody CHAID roste s počtem kategorií závisle proměnné a prediktorů. Použitím Bonferroniho korekce je možné zmírnit vliv mnohonásobného testování a získat porovnatelné *p* hodnoty pro jednotlivé prediktory s různým počtem kategorií.

Výsledná *p* hodnota pro kontingenční tabulku kategorií závisle proměnné a optimálně sloučeného prediktoru je vynásobena *B* koeficientem, čímž získáme adjustovanou *p* hodnotu pro daný prediktor.

Koeficient *B* je různý pro ordinální a kategoriální proměnnou, neboť u ordinální proměnné může docházet pouze ke slučování sousedních kategorií (potřebujeme tedy provést méně testů), zatímco u kategoriální je testováno slučování všech možných kombinací.

Koeficient *B* pro ordinální proměnnou:

$$B_{ordinal} = \binom{s-1}{r-1}, \tag{3.3}$$

koeficient *B* pro kategoriální proměnnou:

$$B_{kategorial} = \sum_{i=0}^{r} (-1)^i \frac{(r-i)^s}{i!(r-i)!}, \tag{3.4}$$

kde *r* je počet řádků a *s* je počet sloupců kontingenční tabulky kategorií závisle proměnné a optimálně sloučeného prediktoru.

Růst stromu se zastaví, pokud je dosaženo následujících pravidel:

- není možné nalézt žádné významné rozdělení.
- Všechna pozorování závisle proměnné v uzlu mají stejnou hodnotu nebo identickou hodnotu pro každý prediktor.
- Pokud je dosaženo uživatelem definovaných nastavení, která se týkají:
    a. parametrů velikosti stromu jako je nastavení počtu terminálních uzlů nebo větví;

b. počtu pozorování v uzlu, které je menší než minimum stanovené uživatelem nebo počtu pozorování, které by po rozdělení vedlo k dceřiným uzlům s menším počtem pozorování, než je definováno uživatelem.

Celkovou správnost stromu OA_{kateg} určujeme stejně jako v případě stromu CART. K odhadu obecné chyby $e(t)$ je možné opět použít k-testovacích souborů z krosvalidace.

Příklad V: Strom typu CHAID

Ukázkový příklad ke cvičení v programu R.

Datový soubor obsahuje informace o pasažérech lodi Titanic, která se potopila v roce 1912 na cestě ze Southamptonu do New Yorku čtyři dny po vyplutí, když narazila do ledovce. Na palubě bylo přes 2200 pasažérů a členů posádky a přežila pouze necelá třetina cestujících. Informace byly shromážděny Britskou obchodní komorou *(British Board of Trade*) při šetření potopení lodi. Celkem 2202 shromážděných záznamů se týkají třídy, kterou cestovali (první, druhá a třetí třída a členové posádky), pohlaví, přežití a věku (pouze rozdělení na dospělé a děti) pasažérů. Je dobře známým faktem, že zejména u přežití žen a dětí hrála důležitou roli třída, kterou cestovaly [13].

```
> library(MASS)
> data(Titanic)
> summary(Titanic)
         class           age          gender         survival
 Crew        :885   Adult   :2092   Female: 470   Missing :1490
 First Class :325   Children: 109   Male  :1731   Survival: 711
 Second Class:285
 Third Class :706
```

Knihovna pro výpočet stromu CHAID není ve standardní nabídce instalace R[6], je však možné ji nainstalovat pomocí příkazu[7]:

```
> install.packages("CHAID", repos="http://R-Forge.R-project.org")
```

Současný algoritmus umožňuje použití pouze kategoriálních a ordinálních proměnných. Spojitá data musí být před použitím převedena na ordinální.

Skript obsahuje dvě části. První je pro nastavení parametrů algoritmu *chaid_control*. Zde je uvedeno defaultní nastavení:

```
> chaid_control(alpha2 = 0.05, alpha3 = -1, alpha4 = 0.05,
              minsplit = 20, minbucket = 7, minprob = 0.01)
```

Parametr *alpha2* určuje hladinu významnosti použitou pro slučování kategorií prediktoru (krok 3), *alpha3* – pokud je zadána kladná hodnota < 1 je hladina významnosti použita také pro rozdělení již dříve sloučených kategorií prediktoru (krok 4), jinak je tento krok vypuštěn (defaultní nastavení), *alpha4* určuje hladinu mezní významnosti pro adjustovanou hodnotu prediktoru (krok 6), *minsplit* definuje minimální počet pozorování, při němž ještě dojde k dalšímu rozdělení uzlu, *minbucket* je minimální počet pozorování v potenciálním terminálním uzlu a *minprob* udává minimální frekvenci pozorování v terminálních uzlech.

[6] V době přípravy skript.
[7] Skript je funkční pro verzi R-2.14.0 a vyšší.

```
> chaid(formula, data, subset, weights, na.action = na.omit,
      control = chaid_control())
```

Funkce *chaid* spouští výpočet stromu a lze u ní nastavit další parametry týkající se souboru jako je: *subset* pro definování testovacího souboru (je-li k dispozici), *weights* pro nastavení vah u jednotlivých pozorování, parametr *na.action*, který určuje, jak bude naloženo s nevyplněnými hodnotami (defaultní nastavení je odstranění řádků s prázdnou hodnotou z výpočtu) a *control*, který definuje parametry algoritmu popsané výše.

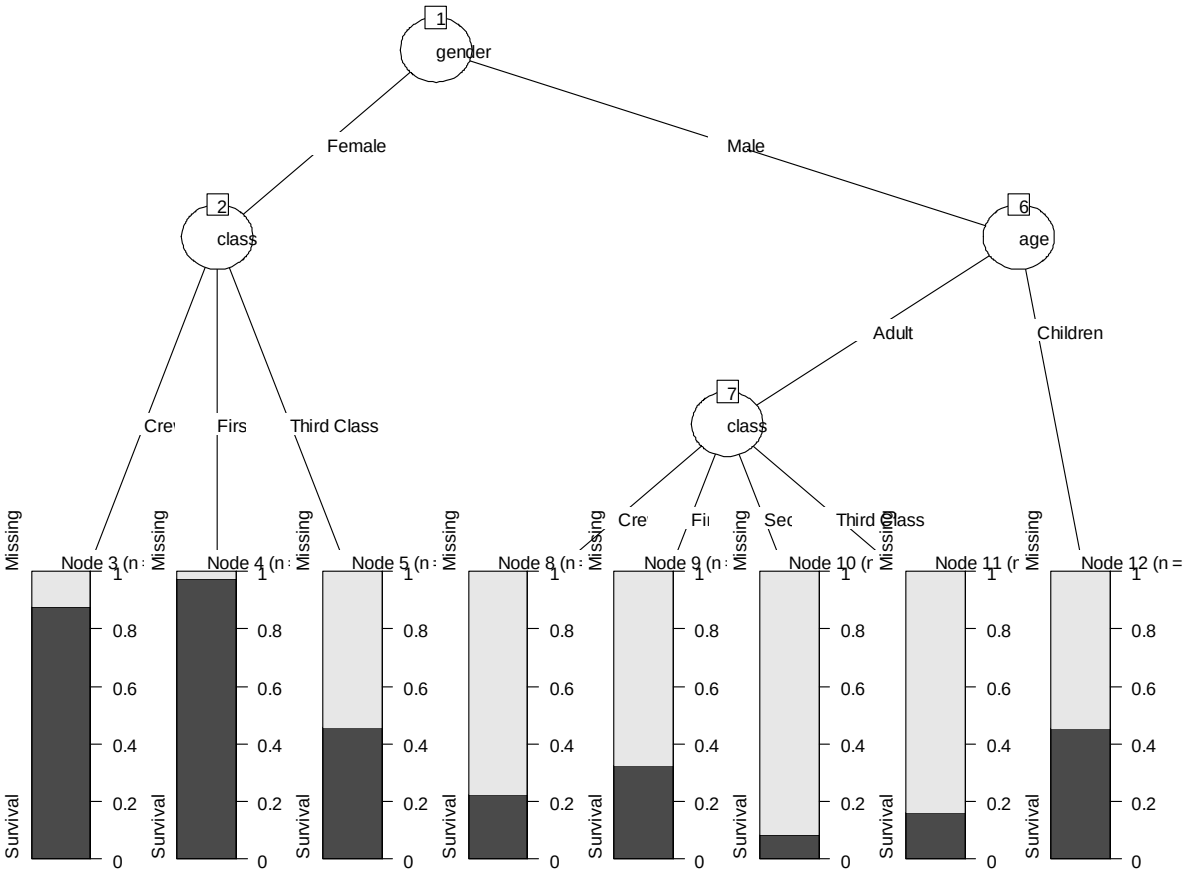
```
> library("CHAID")
> set.seed(123)
> ctrl <- chaid_control(minsplit = 200, minprob = 0.1, alpha2 = 0.05, alpha3
= -1, alpha4 = 0.05)
> chaidTitanic <- chaid(survival ~ class+age+gender, data = Titanic, control
= ctrl)
```

Výsledky v textové podobě zobrazíme pomocí funkce *print*. Hodnota α pro sloučení i výsledné testování sloučených kategorií byla nastavena na 0,05. S touto hladinou významnosti byly porovnávány *p* hodnoty kontingenčních tabulek přežití versus věk, pohlaví a třída.

```
> print(chaidTitanic)

Model formula:
survival ~ class + age + gender

Fitted party:
[1] root
|   [2] gender in Female
|   |   [3] class in Crew, Second Class: Survival (n = 129, err = 12.4%)
|   |   [4] class in First Class: Survival (n = 145, err = 2.8%)
|   |   [5] class in Third Class: Missing (n = 196, err = 45.9%)
|   [6] gender in Male
|   |   [7] age in Adult
|   |   |   [8] class in Crew: Missing (n = 862, err = 22.3%)
|   |   |   [9] class in First Class: Missing (n = 175, err = 32.6%)
|   |   |   [10] class in Second Class: Missing (n = 168, err = 8.3%)
|   |   |   [11] class in Third Class: Missing (n = 462, err = 16.2%)
|   |   [12] age in Children: Missing (n = 64, err = 45.3%)

Number of inner nodes:    4
Number of terminal nodes: 8
```

Výsledkem je, podobně jako v případě stromu typu CART, hierarchická textová podoba stromu, kdy je pro každý uzel uvedena hodnota kategorie prediktoru. Pro terminální uzel je dále zobrazena kategorie závisle proměnné, počet pozorování v uzlu a klasifikační chyba. Výsledná hodnota terminálního uzlu (přežil/nepřežil) je určena jako převládající kategorie závisle proměnné v tomto uzlu a klasifikační chyba je procento chybně klasifikovaných pozorování.

```
> plot(chaidTitanic,cex=0.6)
```



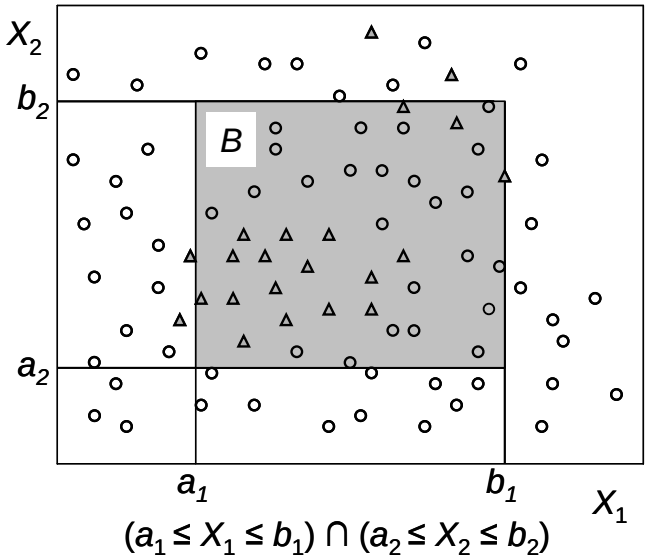
Z výsledků vyplývá, že největší vliv na přežití mělo pohlaví pasažérů (první rozdělení ve stromu), výrazně lépe na tom byly ženy. U žen (a dívek, protože jsou mezi nimi zahrnuty i děti) byl však rozdíl v třídě. Více než 50% žen a dívek ve třetí třídě katastrofu nepřežilo, zatímco v první třídě přežily téměř všechny. Tento rozdíl je patrný i u mužů, ale není tak výrazný (30% vs. 16% mezi 1. a 3. třídou). Vyšší přežití, bez ohledu na třídu, měli v této větvi stromu chlapci.

Dále může čtenář otestovat, zdali dojde při různém nastavení α (například volbou přísnější hranice 0,01) a počtu pozorování v terminálních uzlech ke změně výsledného stromu. Všimněme si také, že byl vynechán krok 4 (možné rozdělení již sloučených kategorií) nastavením parametru *alpha3* na zápornou hodnotu.

3.2 PRIM (*Patient Rule Induction Method*)

Metoda PRIM [14] je primárně určena pro regresní problémy, závisle proměnná je tedy spojitá.

PRIM podobně jako ostatní rozhodovací stromy rozděluje pozorování závisle proměnné Y pomocí hodnot prediktorů do uzlů $t_1,..,t_N$, zde označovaných jako okna $B_1,\ldots, B_K$. Graficky můžeme okna znázornit jako jednotlivé regiony v prostoru prediktorů $X_1,\ldots, X_M$. V případě metody PRIM se však vyhledávají takové regiony, ve kterých je průměr hodnot závisle proměnné Y nejvyšší (nebo nejnižší). Výsledkem je sada jednoduchých pravidel, která definují jednotlivá okna a rozdělují pozorování závisle proměnné (obr. 3.2).



Obr. 3.2 *Mějme 100 pozorování. Závisle proměnná Y označuje presenci $y_i = 1$ (trojúhelníky) nebo absenci $y_i = 0$ (kolečka) určitého druhu rostliny. Pro jednoduchost uvažujme pouze dva spojité prediktory: teplotu X_1 a srážky X_2. Rostlina bude přítomna s větší pravděpodobností v podmínkách daných rozsahem prediktorů $(a_1 \le X_1 \le b_1) \cap (a_2 \le X_2 \le b_2)$, které jsou zde znázorněny pomocí okna B.*

PRIM je tedy velmi vhodný pro případy, kdy nás zajímá nalezení skupin v datech s nejvyšší nebo nejnižší hodnotou závisle proměnné. Například pro různá ochranná opatření, kdy výsledky mohou sloužit ke stanovení vhodné velikosti území podle pravděpodobnosti výskytu druhu nebo ke zjištění klimatických podmínek, při kterých dochází k největšímu znečištění ovzduší, jak si ukážeme v příkladu VI.

Algoritmus stromu postupuje hierarchicky. Začíná se s oknem obsahujícím všechna trénovací pozorování. Okno se postupně zmenšuje o předem zadané procento pozorování. Zmenšování probíhá posunutím vždy pouze jedné hrany okna. Vybere se takové „ořezání“, které způsobí nejvyšší průměr hodnot závisle proměnné Y ve zmenšeném okně. To se opakuje do předem definované hodnoty minimálního počtu pozorování (např. 10) v okně. Následně se okno zpětně zvětšuje podél některé z hran tak dlouho, dokud průměr závisle proměnné uvnitř zvětšovaného okna roste. Těmito kroky získáme sekvenci oken s různým počtem pozorování. Na základě krosvalidace se vybere okno optimální velikosti, pozorování z tohoto okna se odstraní a procedura začíná znovu se zmenšeným souborem.

Algoritmus růstu stromu PRIM

1. Soubor se rozdělí na testovací a trénovací (v poměru zadaném uživatelem).
2. Seřadí se hodnoty prediktorů od nejmenší po největší.
3. Na začátku obsahuje okno celý trénovací soubor.
4. Okno se zmenšuje vždy podél jedné hrany (prediktoru) o malé množství pozorování (často o 5% nebo 10%).
5. Vybere se hrana, pro kterou byl výsledný průměr proměnné *Y* ve zmenšeném okně největší a původní okno se zmenší podél této hrany, tzn. z původního okna jsou odstraněna pozorování s nejnižší/nejvyšší hodnotou prediktoru *X*.

Krok 4 a 5 se opakuje, dokud okno neobsahuje předem stanovené minimum pozorování.

6. Dochází k reverznímu procesu - okno je zpětně rozšiřováno do všech směrů, ale jen pokud se zvýší průměr hodnot *Y* v okně. Opět je zvolen podíl pozorování pro zvětšování okna (např. $\alpha = 0{,}1$).

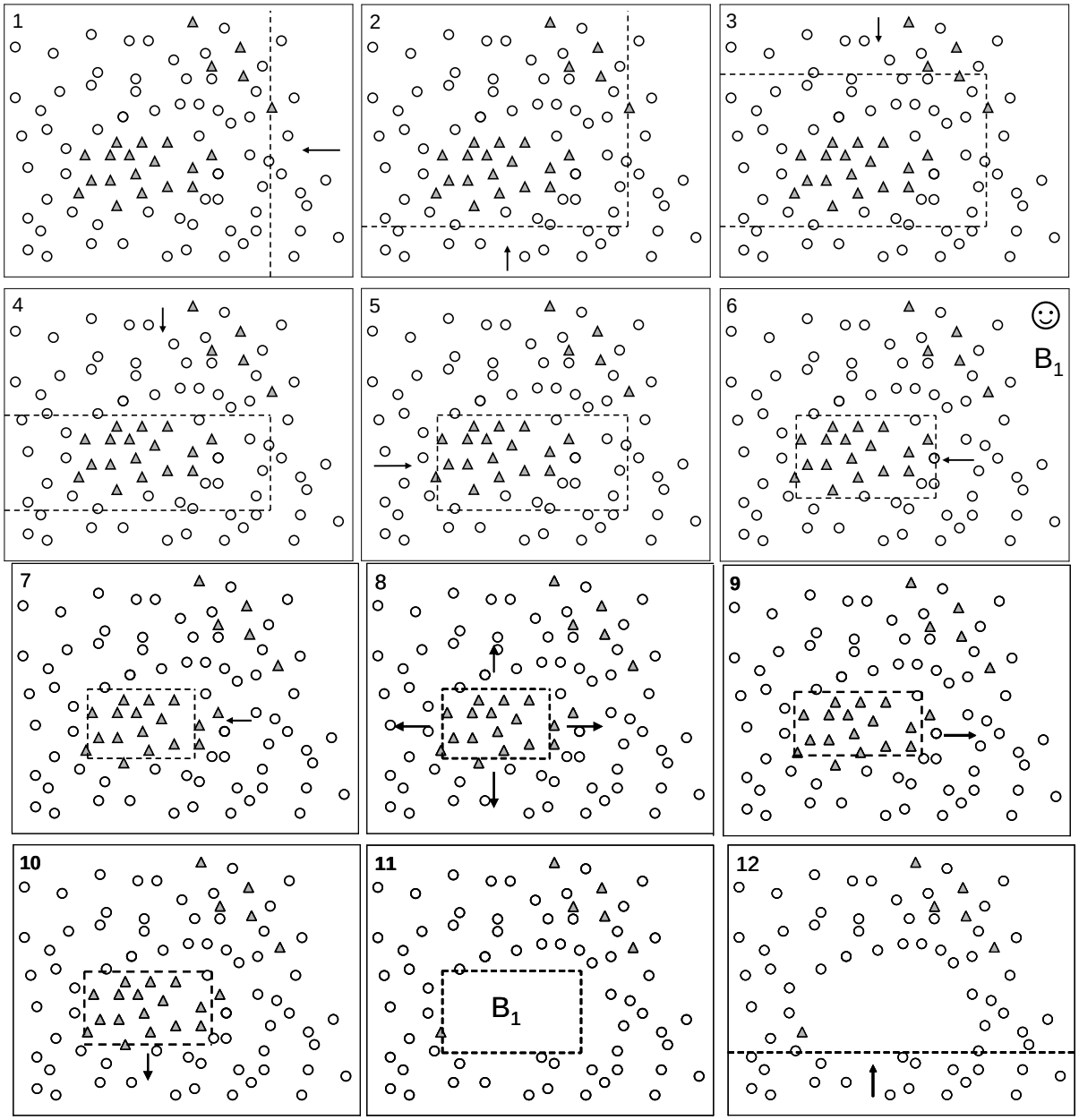
Z kroku 4-6 získáme sekvenci oken o různém počtu pozorování.

7. Vybere se optimální velikost okna pomocí krosvalidace.
8. Pozorování z okna vybraného krosvalidací (označme jej B_1) jsou odstraněna ze souboru.

Se souborem zmenšeným o pozorování z B_1 se algoritmus vrací zpět do kroku 3.
Krok 4-8 se opakuje, dokud není dosaženo konečného počtu oken $B_1, B_2, \ldots, B_K$.

Okna $B_1, B_2, \ldots, B_K$ tvoří výslednou sadu rozhodovacích pravidel. Jednotlivé kroky algoritmu jsou zobrazeny na obrázku 3.3.

Všimněme si, že v kroku 8 je opět použita krosvalidace, podobně jako tomu bylo u předchozích algoritmů. Ve chvíli, kdy jsme zmenšováním a následným zvětšováním získali okna různé velikosti, bylo potřeba vybrat optimální okno. Zřejmou volbou by bylo vybrat okno s nejvyšší hodnotou průměru proměnné *Y*. Mohlo by však dojít k přetrénování modelu, protože nejvyšší hodnotu průměru proměnné *Y* by výsledné okno mohlo mít pouze pro data, na kterých probíhalo učení modelu. Riziko navíc narůstá se zmenšujícím se počtem pozorování v okně. Rozhodovací pravidla pro okno s nejvyšším průměrem na trénovacím souboru, které by navíc mělo pravděpodobně málo pozorování, tak nebudou objektivní. Výběr vhodného okna tedy probíhá na testovacím souboru. Je vybráno okno, které má nevyšší průměr hodnot závisle proměnné pro testovací pozorování.



Obr. 3.3 *Algoritmus metody PRIM. Závisle proměnná Y má hodnotu 1 (trojúhelníky) pokud je 0,2 < X_1 < 0,8 a 0,4 < X_2 < 0,9, jinak 0 (kolečka). Proporce bodů, o které se hrana okna posune, je α = 0,1 (10%). Šipky označují posunutí hrany v daném okně. První fáze algoritmu, kdy dochází ke zmenšování oken* ***1-7****, končí v okně* ***7****, kdy už zmenšování nevede k nárůstu průměru hodnot Y v okně (krok 4 a 5). Následně dochází ke zpětnému rozšiřování okna* ***8 - 10*** *(krok 6). Na základě krosvalidace je vybráno okno* ***6*** *(krok 7), pozorování z tohoto okna jsou odstraněna* ***11*** *(krok 8) a začíná se znovu oknem* ***12*** *(krok 4) [6].*

Stejně jako u ostatních rozhodovacích stromů lze použít kategoriální prediktor, rozhodovací pravidla jsou pak dána podmnožinou kategorií prediktoru. Oproti algoritmu CART je výhodou, že se probere větší škála pravidel a můžeme snadněji najít optimální řešení, než při použití proměnné pouze na základě nejlepší hodnoty kriteriální statistiky. Hlavní nevýhodou je absence stromové struktury, výsledkem je pouze sada pravidel.

Příklad VI: Strom typu PRIM

Ukázkový příklad ke cvičení v programu R.

Podíváme se na stejný příklad, jaký byl použit pro regresní stromy: závislost koncentrace ozónu (ppb) na teplotě (stupně Fahrenheita), rychlosti větru (míle/h) a intenzitě slunečního záření (cal/cm^2). Soubor obsahuje 111 měření [7].

Načteme knihovnu *lattice*, která obsahuje výše popsaný datový soubor se jménem *environmental* a knihovnu *prim* s funkcemi pro výpočet:

```
> library(prim)
> library(lattice)
> data(environmental)
```

Do proměnné *Y* uložíme závisle proměnnou koncentraci ozónu, do proměnné *X* prediktory teplotu a rychlost větru:

```
>  y <-environmental[,1]
>  x <-environmental[,3:4]
```

Připomeňme základní popisnou statistiku souboru:

```
> summary(environmental)
      ozone          radiation       temperature        wind      
 Min.   :  1.0   Min.   :  7.0   Min.   :57.0   Min.   : 2.300  
 1st Qu.: 18.0   1st Qu.:113.5   1st Qu.:71.0   1st Qu.: 7.400  
 Median : 31.0   Median :207.0   Median :79.0   Median : 9.700  
 Mean   : 42.1   Mean   :184.8   Mean   :77.8   Mean   : 9.939  
 3rd Qu.: 62.0   3rd Qu.:255.5   3rd Qu.:84.5   3rd Qu.:11.500  
 Max.   :168.0   Max.   :334.0   Max.   :97.0   Max.   :20.700  
```

U funkce *prim* lze nastavit různé parametry: *peel.alpha* a *paste.alpha* určují podíl pozorování, o které se okno bude zmenšovat, respektive zvětšovat; *mass.min* je minimální podíl pozorování z celkového souboru (defaultně 0,05). Za diskriminační hladinu proměnné *Y* je používán průměr, pokud je parametr *threshold.type* roven 1, je hledáno okno s hodnotami závisle proměnné >= než průměr, při nastavení na nulu hledáme hodnoty <= průměru. Nastavením *threshold.type=0* můžeme zvolit rozsah hodnot závisle proměnné.

Výsledky výpočtu uložíme do objektu *prim.ozon*:

```
> prim.ozon <- prim(x , y = y, threshold.type = 1)
```

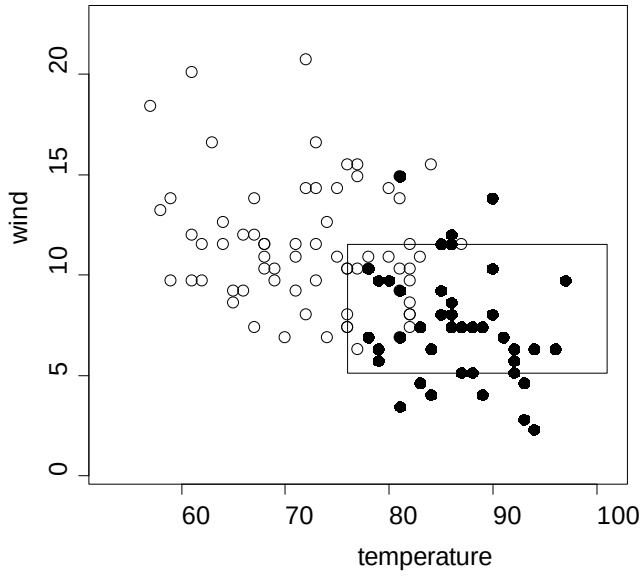
Níže jsou zobrazeny výsledky pro koncentraci ozónu. Sloupeček *box-mean* obsahuje průměrnou hodnotu závisle proměnné v okně a *box-mass* podíl pozorování v okně z celkového souboru. Hvězdička označuje „zbytek“ datového souboru, který již nebyl použit pro rozdělení. Následují pravidla, která definují rozsah okna.

Vyšší koncentrace ozónu 53,5 ppb (box1) nastává při vyšších teplotách v rozsahu 76-101°F a menší rychlosti větru v rozsahu 5,1 – 11,5 (míle/h). Okno obsahuje 47,7 % vzorků. Průměrná koncentrace ozónu v okně definovaná těmito pravidly je však pouze lehce nad celkovým průměrem.

```
> summary(prim.ozon, print.box = TRUE)

         box-mean  box-mass threshold.type
box1    53.50943 0.4774775              1
box2*   31.67241 0.5225225             NA
overall 42.09910 1.0000000             NA

Box limits for box1
    temperature wind
min          76  5.1
max         101 11.5

Box limits for box2
    temperature  wind
min          53  0.46
max         101 22.54
```

Zobrazíme výsledný graf a jednotlivá měření. Plná kolečka obsahují hodnoty koncentrace, které jsou vyšší než průměr. Vidíme, že většina z nich se vyskytuje ve výsledném okně. Prázdná kolečka pak zobrazují pozorování s hodnotami ozónu menšími než celkový průměr.

```
> plot(prim.ozon, col = "transparent",cex.axis=1.5, cex=1.5, cex.lab=1.5)
> points(x[y > 42.1, ], pch=16, cex=1.5)
> points(x[y < 42.1, ], cex=1.5)
```



Ke stávajícím prediktorům přidáme ještě intenzitu slunečního záření.

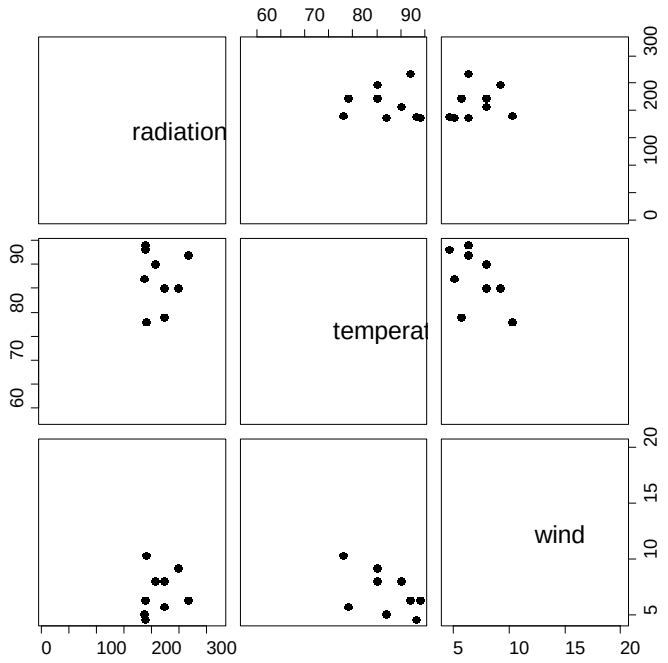
```
>  x <-environmental[,2:4]
> prim.ozon1 <- prim(x , y = y, threshold.type = 1)
> summary(prim.ozon1, print.box = TRUE)

         box-mean  box-mass threshold.type
box1    79.26667 0.2702703              1
box2*   28.33333 0.7297297             NA
overall 42.09910 1.0000000             NA

Box limits for box1
    radiation temperature  wind
min    166.50        78.0  0.46
```

```
max    271.25        96.1 10.87

Box limits for box2
    radiation temperature  wind
min     -25.7          53  0.46
max     366.7         101 22.54
```

Opět jsme získali dvě okna, nyní je ve výsledném okně pouze 27% ze všech pozorování. Hodnoty průměrné koncentrace jsou však mnohem vyšší. Změnily se rovněž rozsahy hodnot teploty a větru. Rozsah hodnot teplot je menší (78-96,1°F), zatímco rozsah hodnot rychlostí větru se zvětšil (0,5-10,9 míle/h). Nejvyšší hodnoty koncentrace ozónu (79,3 ppb) jsou při vyšší teplotě a radiaci a nízké rychlosti větru, což koresponduje s výsledky regresního stromu (Příklad III).

Výsledný graf zobrazuje pozorování s vyššími koncentracemi (box1) v prostoru tří prediktorů.

```
> plot(prim.ozon1, cex=1.5, pch=16,cex.axis=1.5, cex.lab=1.5, col='black')
```



3.3 MARS (*Multivariate Adaptive Regression Splines*)

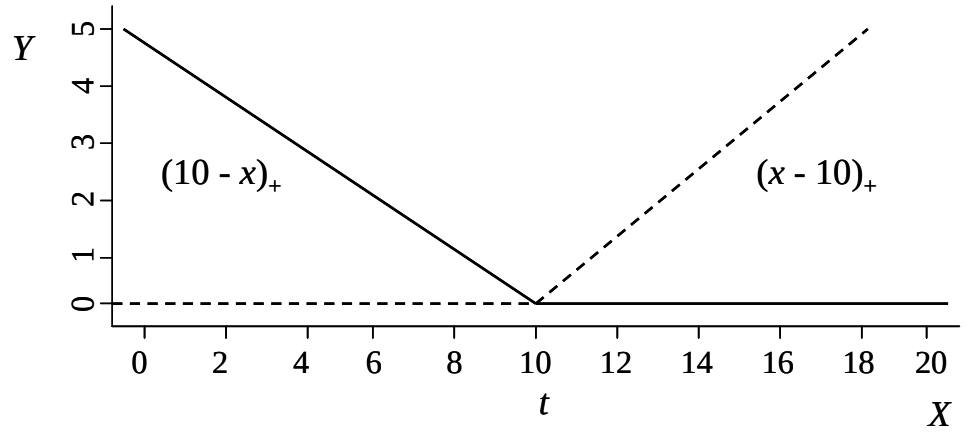
Metoda MARS [15] se nachází na rozhraní mezi stromovou technikou a parametrickou regresí. Je určena pro regresní problémy a odstraňuje určité nedostatky binárních regresních stromů, především nespojitosti odhadnutých hodnot závisle proměnné (připomeňme, že finální počet predikovaných hodnot je roven počtu terminálních uzlů, viz kap. 2.1). Prediktory mohou být spojité i kategoriální. Výsledkem metody je regresní rovnice, chybí tedy klasická stromová struktura a interpretace výsledků při velkém počtu proměnných může tak být obtížnější[8].

Zatímco u ostatních rozhodovacích stromů jsme si pro rozdělení uzlu vystačili s hodnotou prediktoru, který byl určen pomocí kriteriální statistiky (např. druh se bude vyskytovat, pokud je nadmořská výška větší než 200 m.n.m., jinak nikoli), zde budeme potřebovat polopřímku. K rozdělení pozorování závisle proměnné se tedy nepoužívá konstanta, ale lineární aproximace.

Lineární aproximace je uskutečněna pomocí tzv. lineárních splajnů[9] (*linear splines*). Splajny definované v metodě MARS jsou po částech lineárních funkce $(x - t)_+$ a $(t - x)_+$ s uzlem v bodě t, kde + označuje kladnou část funkce. Dvojice těchto funkcí jsou označovány jako zrcadlové páry (obr. 3.4) a samotné funkce $(x - t)_+$ a $(t - x)_+$ jako bázové funkce (*basis functions*).

$$(x-t)_+ = \begin{cases} (x-t) & \text{pokud} \quad x > t \\ 0, & \text{jinak} \end{cases} \qquad (t-x)_+ = \begin{cases} (t-x) & \text{pokud} \quad x < t \\ 0, & \text{jinak} \end{cases} \tag{3.5}$$

Alternativní zápis, který se často používá, je $max(0, x - t)$ a $max(0, t - x)$.



Obr. 3.4 *Příklad funkce zrcadlového páru $(10 - x)_+$ a $(x - 10)_+$, kde $t = 10$ [6].*

Mějme jednoduchou regresní rovnici:

$$Y = \beta_0 + \sum_{m=1}^{M} \beta_m (X_m) + \varepsilon \,, \tag{3.6}$$

[8] K pochopení této kapitoly doporučuji čtenáři zopakovat si základní znalosti pro výpočet lineární regrese.

[9] Nejčastější použití mají splajny v interpolačních úlohách. Pokud prokládáme data polynomem vyššího stupně, dochází často k oscilacím. Místo takového polynomu lze použít funkci, která bude polynomem nízkého stupně a bude prokládat data po částech tak, aby na sebe jednotlivé části navazovaly. Právě tyto funkce se nazývají splajny.

kde Y je závisle proměnná, $X_1,\ldots, X_M$ jsou prediktory, β_0 je intercept a $\beta_1,\ldots, \beta_M$ regresní koeficienty. U jednorozměrné lineární regrese je k vyjádření závislosti Y na X použita přímka a koeficienty jsou odhadnuty metodou nejmenších čtverců.

Nyní předpokládejme model s jedním prediktorem a hodnotou uzlu $t = 10$, který můžeme zapsat pomocí dvou regresních rovnic:

$$\begin{aligned} Y &= \beta_0 + \beta_1(X_1) + \varepsilon \ \text{ pro } \ x > 10 \\ Y &= \beta_0 + \beta_2(X_1) + \varepsilon \ \text{ pro } \ x < 10 \end{aligned} \tag{3.7}$$

S použitím vztahu (3.5) lze rovnice vyjádřit jako:

$$Y = b_0 + b_1(X_1 - t)_+ + b_2(t - X_1)_+ + \varepsilon\,, \tag{3.8}$$

kde $b_0 \equiv \beta_0$, $b_1 \equiv \beta_1$ a $b_2 \equiv \beta_2$.

Stejně jako u lineární regrese lze i u metody MARS použít interakce proměnných.

Pro dva prediktory X_1, X_2:

$$Y = b_0 + b_1(X_1 - t_1)_+ + b_2(t_1 - X_1)_+ + b_3(X_1 - t_1)_+(X_2 - t_2)_+ + \varepsilon\,, \tag{3.9}$$

z čehož plyne:

$$\begin{aligned} &Y = b_0 + b_1X_1 - b_1t_1 + \varepsilon && pro \quad X_1 > t_1\, a \quad X_2 < t_2 \\ &Y = b_0 - b_2X_1 + b_2t_1 + \varepsilon && pro \quad X_1 < t_1 \\ &Y = b_0 + b_1X_1 - b_1t_1 + b_3(X_1X_2 - t_1X_1 - t_2X_1 + t_1t_2) + \varepsilon && pro \quad X_1 > t_1\, a \quad X_2 > t_2 \end{aligned} \tag{3.10}$$

Regresní funkci pro MARS můžeme tedy vyjádřit jako:

$$Y = b_0 + \sum_{m=1}^{M} b_m h_m(X) + \varepsilon\,, \tag{3.11}$$

kde h_m jsou bázové funkce nebo jejich interakce a koeficienty b_m pro dané h_m jsou odhadovány stejně jako u lineární regrese metodou nejmenších čtverců.

Algoritmus je velmi podobný postupnému dopřednému výběru (*forward stepwise selection*) vysvětlujících proměnných v regresním modelu, namísto proměnných se ale vybírají lineární splajny. Začínáme s nulovým modelem (bez prediktorů). Postupně se přidávají jednotlivé členy do rovnice (bázové funkce), ovšem pouze takové, jejichž příspěvek k variabilitě vysvětlené modelem je statisticky významný. Tento příspěvek se určuje na základě snížení residuálního součtu čtverců modelu neboli součtu čtvercových odchylek hodnot y_i závisle proměnné od hodnot $\hat{y}_i$ odhadnutých modelem.

K určení optimální velikosti u metody MARS se používá krosvalidační kritérium *GCV* (*generalized cross-validation*). Na základě *GCV* vybereme model s optimálním počtem členů v rovnici. *GCV* lze použít i pro odhady relativních významností jednotlivých prediktorů.

GCV se určí následovně:

$$GCV(\lambda) = \frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{(1 - M(\lambda)/N)^2}, \tag{3.12}$$

kde N je počet pozorování, $\hat{y}_i$ je hodnota závisle proměnné odhadnutá modelem a $M(\lambda)$ je parametr složitosti modelu, který má tvar:

$$M(\lambda) = r + cK, \tag{3.13}$$

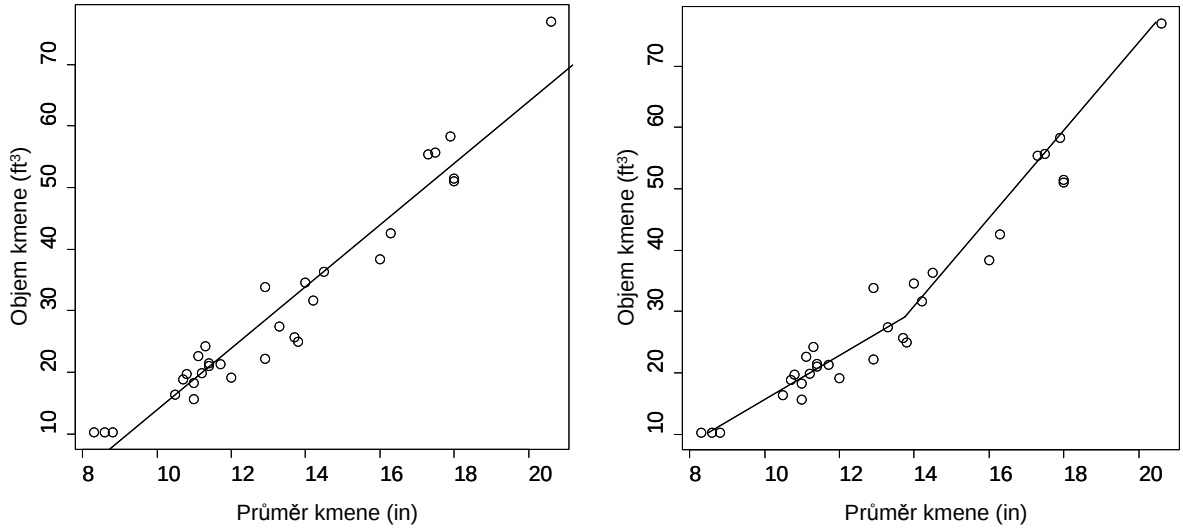
kde r je počet nekonstantních bázových funkcí v modelu a K je počet uzlů t v modelu, kde již proběhl výběr parametrů pomocí dopředného výběru. Konstanta c byla určena experimentálně, $c = 3$ pokud nejsou zahrnuty interakce, $c = 2$ pro rovnici s interakcemi [6]. Datový soubor je rozdělen na testovací a trénovací v poměru zadaném uživatelem (často 70% trénovací a 30% testovací). Na trénovacím souboru je vytvořen model a je spočítána jeho přesnost (R^2) na testovacím souboru. Hodnota *GCV* je spočítána pro různé podmodely, mající různý počet členů v rovnici, který označuje parametr λ. Následně je vybrán podmodel s nejmenší hodnotou *GCV*.

GCV je tedy otestování modelu na nezávislém datovém souboru, adjustované na velikost souboru a počet členů v rovnici. Velmi podobná situace byla u metody CART, kde byl pomocí krosvalidace vybírán optimální počet terminálních uzlů stromu a u metody PRIM, kde se vybírala okna optimální velikosti.

Rozdíl mezi výstupem z metody MARS a lineární regresí si ukážeme na následujícím příkladu.

Příklad VII: Průměr, výška a objem třešní

Datový soubor obsahuje údaje o průměru, výšce a objemu 31 pokácených třešní. Průměr je uveden v palcích a byl změřen ve výšce 4-6 stop. Datový soubor obsahuje 31 pozorování o 3 proměnných [16]. Podíváme se na závislost objemu kmene (v krychlových stopách) na jeho průměru při použití lineární regrese (LR) a metody MARS. Na obrázku 3.5 můžeme porovnat výsledky proložení obou metod.



Obr. 3.5 *Vlevo výsledek proložení LR, vpravo metodou MARS.*

Výsledné rovnice závislosti objemu stromu na jeho průměru pro lineární regresi a MARS:

objem = -36,9 + 5,1 * *průměr*,
objem = 28 + 6,5 * *max*(0; *průměr* - 14) – 3,4 * *max*(0; 14 - *průměr*).

Intuitivně tušíme, že metoda MARS našla v datech „bod zlomu“ neboli hodnotu uzlu *t*. Tím se dostáváme k algoritmu této metody, jejíž hlavní část spočívá v nalezení této hodnoty.

Algoritmus metody MARS

1. Algoritmus začíná s konstantní funkcí $h_m(X) = 1$.
2. Vytvoří se splajny (zrcadlové páry) se svým středem (uzlem *t*) v každé hodnotě x_{ij}, pro každý prediktor X_j. Získáme množinu všech „kandidátských“ bázových funkcí *C* a model je tvořen prvky z této množiny nebo jejich kombinací.
3. Z množiny *C* jsou do modelu přidávány pomocí postupného výběru významné bázové funkce, které snižují reziduální chybu modelu. Proces postupuje hierarchicky, významné interakce jsou přidávány do modelu pouze z kombinace bázových funkcí, které již byly do modelu vybrány.

Z kroku 1 - 3 jsme získali rovnici s vybranými členy. Počet členů však bývá většinou velmi velký a takto sestavený model přetrénovaný.

4. Posledním krokem algoritmu je tedy procedura zpětného odstraňování. Z rovnice jsou odstraněny ty členy, u kterých po jejich odstranění dojde k nejmenšímu zvýšení chyby modelu. Zpětné odstraňování je učiněno pomocí krosvalidace. Hodnota *GCV* je spočítána pro různé velikosti modelu (s různým počtem členů v rovnici) a je vybrán model, pro který je hodnota *GCV* minimální.

Příklad VIII: Strom typu MARS

Ukázkový příklad ke cvičení v programu R.

Vraťme se k příkladu o závislosti objemu kmene 31 třešní na jejich výšce a průměru kmene.

```
> trees
> summary(trees)
     Girth           Height       Volume     
 Min.   : 8.30   Min.   :63   Min.   :10.20  
 1st Qu.:11.05   1st Qu.:72   1st Qu.:19.40  
 Median :12.90   Median :76   Median :24.20  
 Mean   :13.25   Mean   :76   Mean   :30.17  
 3rd Qu.:15.25   3rd Qu.:80   3rd Qu.:37.30  
 Max.   :20.60   Max.   :87   Max.   :77.00  
```

Načteme knihovnu *earth* pro výpočet metody MARS.

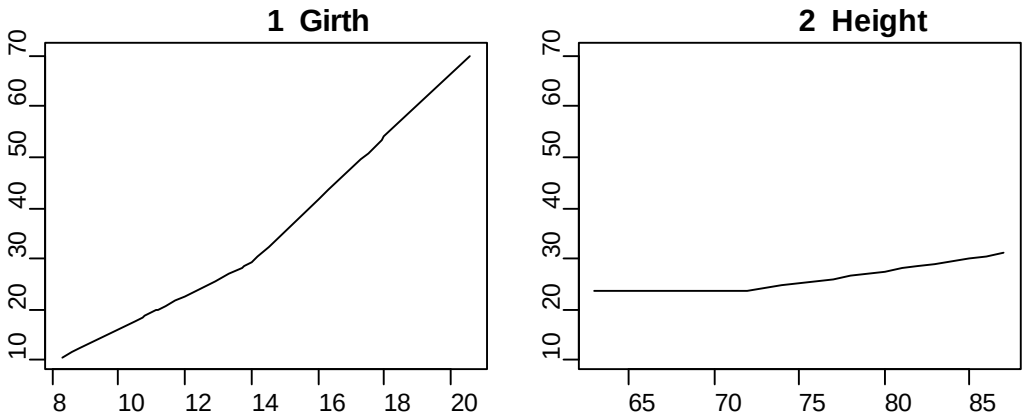
```
> library(earth)
```

U funkce *earth* můžeme nastavit parametry pro výpočet: *nk* specifikuje maximální počet členů v rovnici (včetně interceptu) při postupném výběru členů před tím, než dojde k jejich zpětnému odstraňování pomocí krosvalidace. Defaultní nastavení je určeno z počtu prediktorů a je vhodné jej ověřit pomocí argumentu *trace* (nastavením na jedna). Velmi důležitým parametrem je *degree*, který určuje maximální stupeň interakcí v modelu. Defaultně je nastaven na 1, což znamená aditivní model bez interakcí. Dalším parametrem je *penalty*, který se rovná konstantě *c* použité při krosvalidaci a je nastaven na 3, pokud nejsou zahrnuty interakce (*degree* = 1), nebo na 2 pro rovnici s interakcemi (*degree* > 1). Další kritéria specifikují nastavení krosvalidace: *nfold* je počet rozdělení na podsoubory a je roven hodnotě *k*. Defaultně se krosvalidace neprovádí (*nfold* = 0), jestliže je *nfold* > 0, nejprve je vytvořen model na všech pozorováních, následně je vytvořeno *k* modelů a R^2 je měřen na *k* testovacích souborech. Výsledná hodnota R^2 z krosvalidace *cv.rsq* je průměrem všech R^2 na testovacích souborech. Lze nastavit i počet opakování krosvalidace parametrem *ncross*.

```
> tresne <- earth(Volume ~ ., data = trees, degree =1, nfold = 1)
```

Příkazem *plotmo* zobrazíme závislosti jednotlivých prediktorů z metody MARS. Hodnoty jsou mediány prediktorů.

```
> plotmo(tresne)

 grid:    Girth Height
           12.9     76
```



Pomocí *summary* zobrazíme výsledky metody v textové podobě.

```
> summary(tresne)
Call: earth(formula=Volume~., data=trees, nfold=1, degree=1)

             coefficients
(Intercept)    27.2458996
h(Girth-14)     6.1766918
h(14-Girth)    -3.2662277
h(Height-72)    0.4912072

Selected 4 of 6 terms, and 2 of 2 predictors
Importance: Girth, Height
Number of terms at each degree of interaction: 1 3 (additive model)
GCV 10.60632    RSS 197.0722    GRSq 0.9620131    RSq 0.9756884
```

Výstup z funkce *earth* nás informuje, že byly vybrány čtyři členy z šesti možných a oba prediktory. Model obsahuje dva uzly *t* s hodnotami 14 palců pro průměr kmene a 72 stop pro výšku stromu. Vidíme, že zrcadlový pár pro výšku stromu byl odstraněn. Rovnici můžeme pomocí koeficientů jednoduše přepsat na: *objem* = 27,2 + 6,2 * *max*(0; *průměr* - 14) – 3,3 * *max*(0; 14 - *průměr*) + 0,5 * *max*(0; *výška* - 72).

Další výstupy určují přesnost modelu. *RSS* je residuální suma čtverců modelu. *Rsq* je koeficient determinace (1-*RSS*/*TSS*). GCV je hodnota krosvalidačního kritéria, na základě kterého byl mezi modely s různým počtem členů vybrán právě model se čtyřmi členy v rovnici. *GRSq* odhaduje predikční sílu modelu, v podstatě hodnotu R^2 při použití GCV. V našem případě výsledky ukazují, že model velmi dobře popisuje závislost objemu kmene na jeho průměru a výšce (R^2 = 0,97) a zároveň by predikce objemu kmene na základě těchto prediktorů byla velmi přesná GR^2 = 0,96. Pro ověření výsledků krosvalidací (zejména pro větší *k*) je však k dispozici velmi malý počet pozorování.

Nastavením parametru *trace* = 1 zjistíme postup při vytváření modelu: v jakém pořadí byly vybrány jednotlivé členy a jak se změnil R^2 po zpětném odstranění některých členů.

```
> tresne <- earth(Volume ~ ., data = trees, degree =1, nfold = 1, trace =
1 )
x is a 31 by 2 matrix: 1=Girth, 2=Height
y is a 31 by 1 matrix: 1=Volume
Forward pass term 1, 2, 4, 6, 8
Reached delta RSq threshold (DeltaRSq 0.000280038 < 0.001) at 7 terms
After forward pass GRSq 0.9372 RSq 0.9774
Prune method "backward" penalty 2 nprune 6: selected 4 of 6 terms, and 2
of 2 predictors
After backward pass GRSq 0.962 RSq 0.9757
```

Nyní se podíváme, jak se změní výsledky modelu, pokud budou zahrnuty i interakce proměnných.

```
> tresne1 <- earth(Volume ~ ., data = trees, degree = 2, nfold = 1)
> summary(tresne1)
Call: earth(formula=Volume~., data=trees, nfold=1, degree=3)

             coefficients
(Intercept)    27.2458996
h(Girth-14)     6.1766918
h(14-Girth)    -3.2662277
h(Height-72)    0.4912072
```

```
Selected 4 of 7 terms, and 2 of 2 predictors
Importance: Girth, Height
Number of terms at each degree of interaction: 1 3 (additive model)
GCV 12.06763    RSS 197.0722    GRSq 0.9567793    RSq 0.9756884
```

Nedošlo k žádné změně výsledného modelu, interakce tedy nejsou významné.

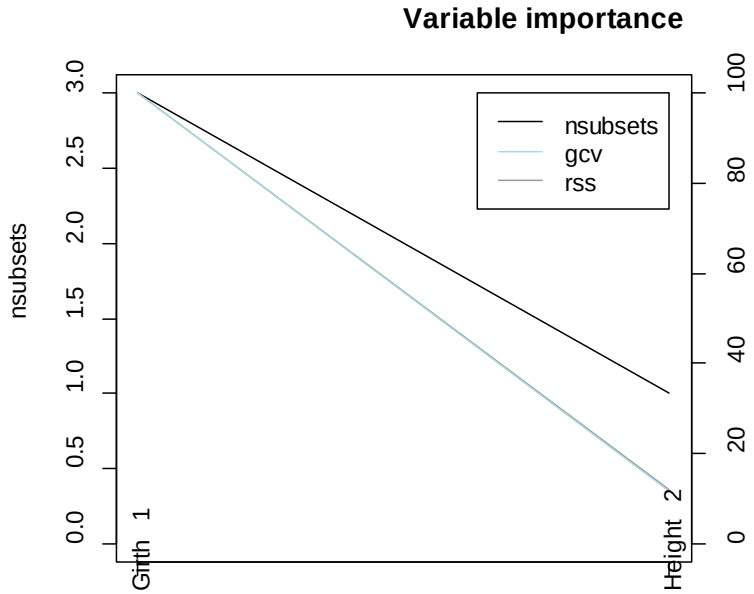
Pomocí funkce *evimp* můžeme spočítat významnost proměnných v modelu na základě jejich příspěvku k vyčerpané variabilitě, přičemž hodnoty jsou standardizovány na 0-100. Parametr *trim* určuje, zda budou zobrazeny výsledky u proměnných, které nebyly vybrány pro žádný podsoubor.

```
> tresne.imp <- evimp(tresne, trim=FALSE, sqrt =TRUE)
> print(tresne.imp)
       nsubsets   gcv    rss
Girth         3 100.0  100.0
Height        1  11.4   12.1
```

Nejvýznamnější proměnnou je objem kmene, výška přispívá k celkové variabilitě podstatně méně. Hodnoty ve sloupečku *nsubsets* označují významnost na základě počtu, kolikrát se proměnná objevila v rovnici při použití různých podsouborů. Ve sloupci *gcv* jsou uvedeny významnosti s použitím GCV kritéria, *rss* je významnost určená použitím celkové residuální sumy čtverců.

Srovnání relativních významností prediktorů zobrazíme v grafu užitím funkce *plot*.

```
> plot(tresne.imp)
```



Další zajímavou funkcí je *predict*, která nám umožní predikci nových pozorování. Protože nemáme k dispozici nová měření, použijeme stejný datový soubor jako pro tvorbu modelu. Zobrazíme prvních deset predikovaných a pozorovaných hodnot.

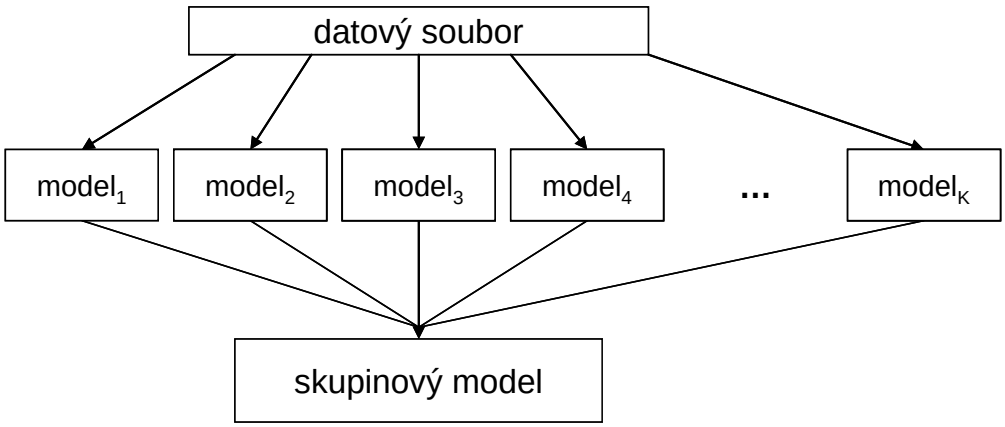
```
> predikovane<- predict(tresne)
> trees[1:10,3]
 [1] 10.3 10.3 10.2 16.4 18.8 19.7 15.6 18.2 22.6 19.9
> predikovane[1:10]
 [1]  8.628402  9.608270 10.261516 15.814103 20.888213 22.197250 17.447216
 [8] 18.920838 21.703497 19.574084
```

Výsledné hodnoty jsou velmi podobné. Můžeme rovněž dosadit vlastní hodnoty pro průměr (10) a výšku (80) třešně a získat předpovězenou hodnotu, která je v tomto případě 18,1 ft^3.

```
> predict(tresne, c(10,80))
       Volume
[1,] 18.11065
```

4 Skupinové modely

Dříve než se dostaneme k jednotlivým typům náhodných lesů, podíváme se na problém trochu obecněji. Lesy totiž patří do skupiny tzv. skupinových (*ensamble*) metod. Princip skupinových modelů je jednoduchý, skupině modelů (např. rozhodovacích stromů) zadáme stejný problém, na kterém se naučí. Výstupy naučených modelů se kombinují (obr. 4.1). Zajímá nás, zda kombinací výsledků modelů dosáhneme zlepšení výsledků predikce či klasifikace. Výsledkem skupinového modelu je zprůměrování všech výsledků jednotlivých modelů v případě regrese nebo většinové hlasování jednotlivých modelů v případě klasifikace (i u klasifikace lze však použít průměrování).



***Obr. 4.1**. Schéma skupinové metody.*

Otázkou je, zdali kombinací modelů můžeme získat přesnější model. Předpokladem pro použití kombinace modelů je podmínka, že jednotlivé modely musejí být různé. Pokud bychom vytvořili sadu modelů na stejném datovém souboru, přidaná hodnota bude nulová, neboť výsledný skupinový model bude shodný s výsledkem jednotlivých modelů. Řešením je použít různé soubory pro učení modelu, které získáme náhodným výběrem z trénovací množiny dat. Modely tak budou vykazovat „odlišné“ chyby. Přesnost a stabilita těchto modelů se následně ověřuje na testovacích souborech.

4.1 Rozklad na systematickou chybu a varianci (*Bias-Variance Decomposition*)

Pojďme se detailněji podívat, čím vším je vlastně chybovost modelu způsobena. Začněme jednoduchým případem, kdy budeme měřit náhodnou veličinu Y v populaci (např. váha člověka) a chceme vyjádřit její reprezentativní hodnotu pro celou populaci. Hledáme takový odhad $\hat{y}$, který minimalizuje střední hodnotu chyby $E_y(y-\hat{y})^2$ přes celou populaci. V ideálním případě bychom změřili všechny vzorky v populaci (zvážili všechny lidi) a zjistili jejich střední hodnotu $E_y(y)$ (např. průměr, medián), kterou bychom prohlásili za optimální odhad. V praxi však tento přístup není možný a pomůžeme si výběrem pouze určité skupiny pozorování z populace, který však musí mít stejné vlastnosti jako celá populace. Takovýto výběr vytvoříme náhodným výběrem.

Stejný případ nastává u modelů, kdy vybíráme pozorování pro trénovací soubor z množiny všech pozorování. Odchylky pozorovaných od predikovaných hodnot (přesnost klasifikace či

regrese) nebudou způsobeny pouze „přírodní“ variabilitou, kterou jsme modelem nevysvětlili, ale také rozdílem ve výsledcích pro různé náhodné výběry a celou populaci.

Mějme soubor trénovacích dat:

$L = (\boldsymbol{y_i},\boldsymbol{x_i}),\ i = 1,\ldots,n.$

Chceme najít takovou funkci v prostoru všech prediktorů a hodnot závisle proměnné, aby predikční chyba byla malá.

Pokud mají (Y,X) stejné rozdělení a daná funkce R udává rozdíl mezi pozorovanou hodnotou y_i a predikovanou hodnotou $\hat{y}_i$ závisle proměnné Y, pak můžeme predikční chybu (*prediction error*) obecně vyjádřit jako:

$$PE(f,L) = E_{Y,X}\, R\big(Y, f(X,L)\big)^2, \tag{4.1}$$

kde $f(X,L)$ jsou predikované hodnoty $\hat{y}_i$ pro trénovací soubor L. Obvykle je závisle proměnná Y jednorozměrná.

Rozdělení chyby na jednotlivé složky modelu si předvedeme na regresi.
Mějme jednoduchou závislost náhodné veličiny Y na náhodném vektoru X:

$$Y = f(X) + \varepsilon, \tag{4.2}$$

kde $f(X) = E(X|Y), \qquad E(\varepsilon|X) = 0$.

Y můžeme rozložit na její strukturální část $f(X)$, která je predikována pomocí X a šum ε, který nejsme schopni vysvětlit pomocí modelu.

Průměrná obecná chyba (*mean-squared generalization error*) na trénovacím souboru L je rovna:

$$PE(f,L) = E_{Y,X}\big(Y - f(X,L)\big)^2 \tag{4.3}$$

Optimální model by měl mít minimální průměrnou chybu pro různé výběry L, což v podstatě znamená, že výsledky modelu pro jednotlivé výběry trénovacích souborů by se neměly příliš lišit.

Vyjádříme průměr trénovacích souborů stejné velikosti ze stejného rozložení:

$$\bar{f}(x) = E_L\, f(x,L), \tag{4.4}$$

kde $E_L\, f(x,L)$ je průměr přes všechny trénovací soubory L predikované hodnoty y_i v hodnotě x_i.

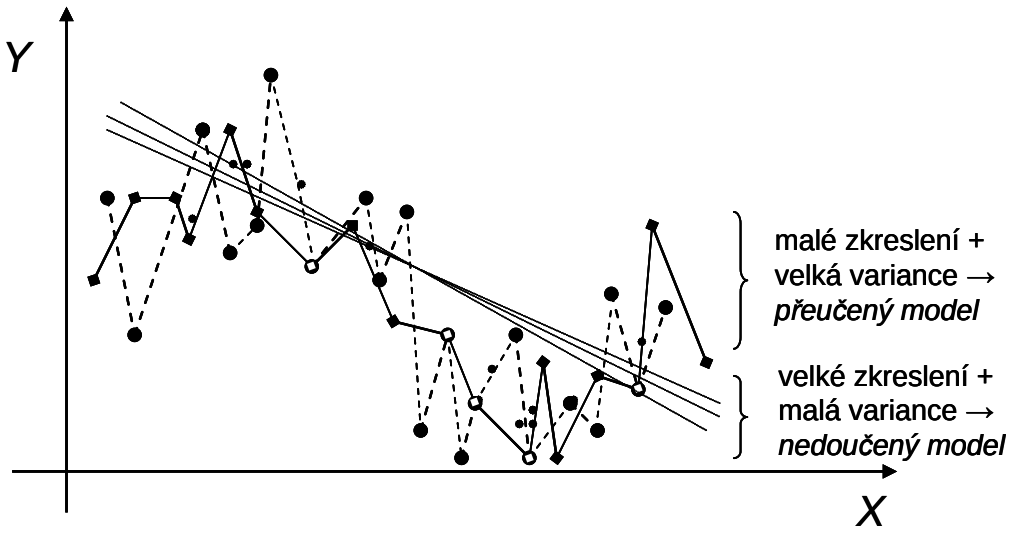
Celkovou chybu modelu můžeme rozdělit na tři složky a to šum, zkreslení (*bias*) a varianci:

$$PE = \underbrace{E\varepsilon^2}_{\text{šum}} + \underbrace{E_{Y,X}\left(f(X) - E_L f(X,L)\right)^2}_{\text{zkreslení}^2} + \underbrace{E_{X,L}\left(f(X,L) - E_L f(X,L)\right)^2}_{\text{variance}} \tag{4.5}$$

- **Šum** – je reziduální chyba neboli minimální dosažitelná chyba modelu, kterou nejsme schopni modelem vysvětlit.
- **Zkreslení^2**- určuje systematickou chybu modelu. Je to rozdíl optimálního modelu od průměrného modelu.
- **Variance –** je variabilita výsledků jednotlivých výběrů, jinými slovy, jak moc se predikované hodnoty $\hat{y}_i$ liší v rámci trénovacích podsouborů L. Vysoká variance značí přeučený model.

Modely, které se používají ve skupinových modelech, se označují jako slabé modely neboli *weak learners* (slabý žák, u klasifikace také slabý klasifikátor). **Slabý model** je definován obecně jako model, který má malé zkreslení, ale vysokou varianci. Slabé modely tedy mají velmi vysokou přesnost, ale pouze pro pozorování z trénovacího souboru. Takovýto model je většinou přeučený a nemá obecnou platnost [17].

Příkladem slabých modelů s malým zkreslením, ale vysokou variancí může být interpolace bodů pomocí lineárních splajnů (obr. 4.2). Čárkovanou a plnou čárou jsou znázorněny interpolace bodů pro dva trénovací soubory. Vidíme, že splajny spojují všechny body, ale takovéto proložení bodů nereflektuje variabilitu souboru, kdy jednotlivá pozorování nemusí být přesně změřená. Modely jsou přeučené. Mají vysokou přesnost, ale variance pro výsledky na různých trénovacích souborech je velká. Naopak, je-li zkreslení velké a variance nízká, dostáváme model, který má nízkou přesnost a nevysvětluje dobře závislost v datech. Takový model je nedoučený a na obrázku 4.2 je znázorněn třemi přímkami (tečkovaně), které vznikly pro tři náhodné výběry z množiny trénovacích dat. Vidíme, že variance je nízká, proložení přímkami pro tři soubory dopadlo velmi podobně.

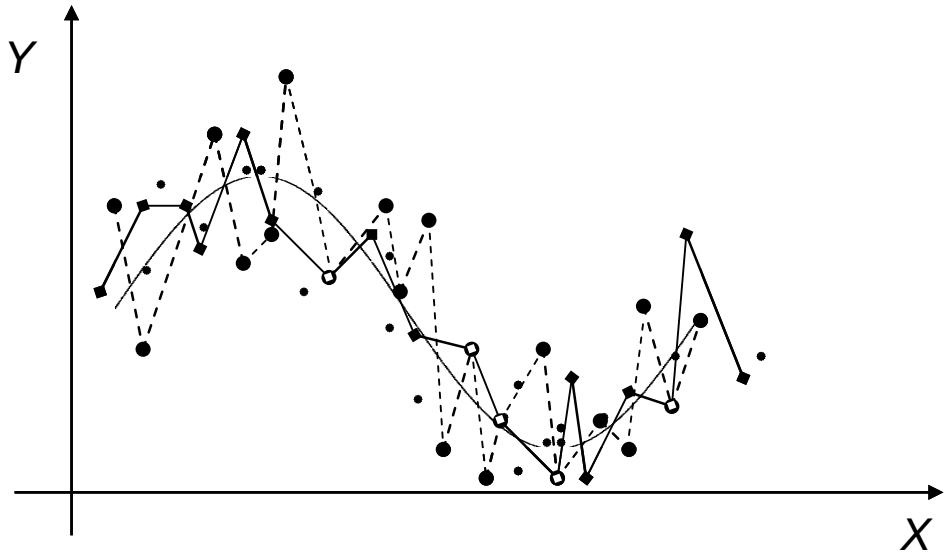


Obr. 4.2 *Ukázka modelu s malým zkreslením a vysokou variancí (proložení bodů lineárními splajny) a modelu s velkým zkreslením a nízkou variancí (proložení přímkami).*

Hledáme tedy model, který by měl nízkou varianci i zkreslení. Kombinováním několika slabých modelů můžeme snížit obě tyto složky.

Jak na to? Vraťme se k příkladu na obrázku 4.3. Vytvořme náhodným výběrem trénovací soubory, které obsahují 2/3 všech pozorování. Všechny body z trénovacího souboru proložíme

pomocí lineárních splajnů. Na obrázku jsou opět znázorněny výsledky pro dva trénovací soubory. Tento postup opakujeme až do předem stanoveného počtu trénovacích souborů (např. 1000). Získali bychom tisíc proložení pomocí splajnů. Následně dojde ke zprůměrování výsledků predikce ze všech slabých modelů, čímž získáme celkový výsledek (na obr. 4.3 jej představuje sinusoida), který má malé zkreslení i varianci.



Obr. 4.3 *Zprůměrováním slabých modelů získáme výsledný model.*

Obecněji:

Mějme *k*-tý slabý model ve tvaru:

$$f_k(x, L) = f(x, L, \Theta_k), \tag{4.6}$$

kde Θ_k je náhodný vektor, na základě kterého se vyberou pozorování pro *k*-tý slabý model. Vektory Θ_k jsou nezávislé, se stejným rozdělením.

Jestliže *N* je celkový počet pozorování trénovacího souboru, každý Θ_k vybere náhodně $2N/3$ pozorování. Hodnoty $y(n)$, $x(n)$ pro vybrané *n*, jsou odstraněny z trénovacího souboru.

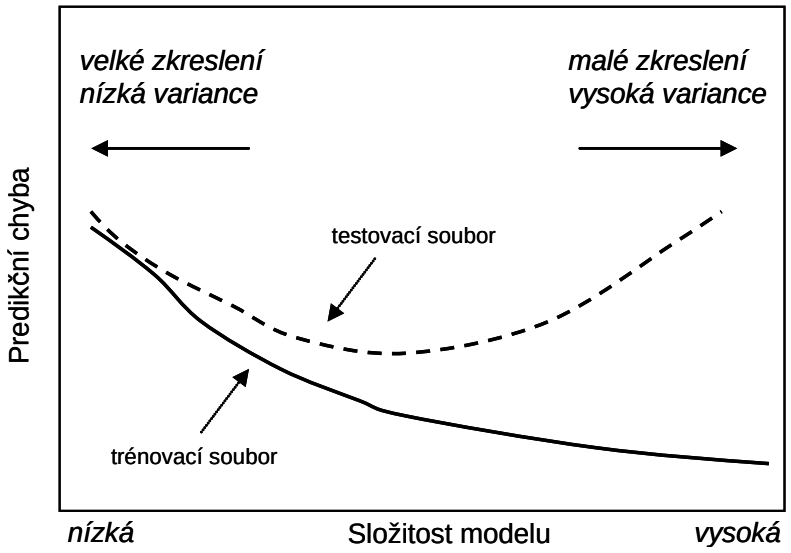
Skupinový model je roven:

$$F(x, L) = \frac{1}{K} \sum_k f(x, L, \Theta_k), \tag{4.7}$$

kde *K* je celkový počet slabých modelů.

Rozhodovací stromy jsou dobrými kandidáty pro použití ve skupinových modelech. Neprořezané stromy mají totiž vysokou přesnost pro trénovací soubor (tedy nízký bias), ale vysokou variabilitu (výsledky mezi testovacím a trénovacím souborem se liší). Rozhodovací stromy, na které nejsou aplikovány metody pro hledání optimální velikosti stromu, jsou tedy podle výše uvedené definice slabými modely.

Připomeňme ještě, že u rozhodovacích stromů jsme pro určení jeho optimální velikosti museli rovněž najít kompromis mezi variancí a zkreslením (obr. 4.4).



Obr. 4.4 *Výběr optimálního stromu pomocí testovacího a trénovacího souboru.*

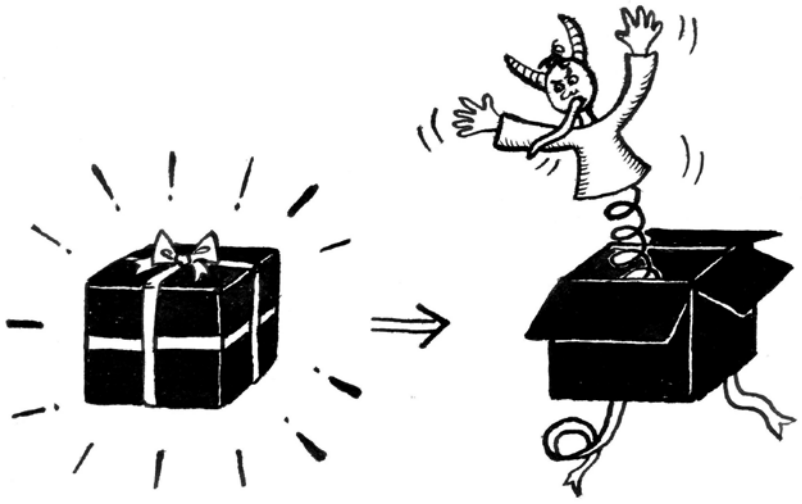
Označení skupinové modely se občas používá také pro kombinaci výsledků z různých modelů (např. neuronových sítí, rozhodovacích stromů a regrese) na stejném souboru.

5 Náhodné lesy pro klasifikaci a regresi

“ looking inside the black box is necessary“
Leo Breiman

V minulé kapitole jsme si ukázali, že kombinací modelů můžeme získat model, který bude obecnější a přesnější než jednotlivé modely. S myšlenkou použití více stromů pro zpřesnění klasifikace a predikce přišel poprvé Breiman [18] a vytvořil tak les. Lesy jsou tedy nadstavbou nad rozhodovacími stromy. Mohou být použity pro klasifikaci i regresi a odstraňují některé problémy, které nastávají při použití stromů, zejména jejich nestabilitu.

Lze si však snadno domyslet, že les je složitější a méně přehledný než jeden strom a informace jednotlivých stromů se ztratí v rámci celého lesa. Někdy je tato metoda díky své „neprůhlednosti“ označována jako černá skříňka (*black box*). Ztráta jednoduchosti však není důvodem k nepochopení pozadí metody.



Existuje několik typů lesů. Nejvíce využívanou metodou v biologii je Random Forest [19]. Je to způsobeno zejména množstvím dalších informací mimo klasifikaci a predikci, které lze z tohoto lesa získat.

5.1 Random Forest

Tato technika byla vyvinuta pro soubory obsahující velké množství prediktorů a velmi dobře funguje i na malých datových souborech. Je hojně používána především v různých genomických a proteomických studiích a pro různé typologie druhových společenstev. Random Forest je určen pro klasifikaci i regresi.

Náhodné lesy lze použít pro celou řadu problémů, kterým se budeme věnovat v následujícím textu, jako jsou:

- klasifikace/predikce;
- měření významnosti proměnných;
- efekt proměnných na predikci;
- shlukování;
- detekce odlehlých hodnot.

Náhodný les se skládá ze souboru stromů $T_1,...,T_N$, jejichž klasifikační nebo regresní funkce lze vyjádřit jako $h(X, \Theta_1),...,h(X, \Theta_N)$, kde h je funkce, X je prediktor a $\Theta_1,...,\Theta_N$ jsou nezávislé stejně rozdělené náhodné vektory. Pro metodu Random Forests se používají binární stromy typu CART. Podobně jako při tvorbě jednotlivých stromů se i zde používá rozdělení na testovací a trénovací soubor. Trénovací soubory pro jednotlivé stromy T_i jsou tzv. **bootstrapové výběry** z datového souboru L. Bootstrapové výběry jsou náhodnými **výběry s opakováním** o velikosti n. Tvorbu výběru s opakováním výběrů si lze jednoduše představit jako losování čísel. Množinou všech pozorování by byla všechna čísla, ze kterých se bude losovat. Při losování se náhodně vytáhne určitý počet čísel. Každé číslo je po jeho vylosování vráceno zpět, může tedy být opět vybráno. Předem stanovený počet vylosovaných čísel tvoří bootstrapový výběr. Po losování jsou všechna čísla vrácena zpět a nový tah (nový výběr) začíná opět se všemi čísly. Takto lze při každém tahu vybrat stejný počet čísel, aniž by se nám snižovala velikost množiny původních pozorování. Tímto způsobem lze rozdělit i velmi malé soubory na velký počet trénovacích a testovacích souborů.

Bootstrapové výběry ale mají jednu nevýhodu, jednotlivé výběry nejsou nezávislé jako například u krosvalidace, neboť do bootstrapového výběru jsou některá pozorování vybrána opakovaně a některá naopak vůbec. Počet pozorování, která se do bootstrapového výběru nedostanou je přibližně 37%[10] [18].

Pozorování, která jsou v i-tém bootstrapovém výběru L_i, se použijí při tvorbě stromu T_i (trénovací soubor), naopak pozorování, která se do toho výběru nedostala (testovací soubor) jsou použita k odhadu jeho chyby. Odhady chyby na testovacím souboru se nazývají *oob* (*out-of-bag, out of bootstrap sample)* odhady. Celkový počet *oob* pozorování tvoří 1/3 datového souboru.

Při použití lesů pro klasifikaci získáme z každého stromu informaci o zařazení každého pozorování do výsledné kategorie. Výsledek klasifikace lesa je dán většinovým hlasováním všech stromů.

$$\hat{C}_{rf} = vetsinove\_hlasovani\left\{\hat{C}_i(x)\right\}_1^N, \tag{5.1}$$

kde $\hat{C}_i(x)$ je výsledek klasifikace i-tého stromu.

Pokud je les použit pro regresi, predikce každého pozorování je jednoduše průměrem ze všech stromů.

$$\hat{f}_{rf}(x) = \frac{1}{N}\left(\sum_{i=1}^{N} T_i(x)\right) \tag{5.2}$$

Náhodný les zvyšuje přesnost (snižuje zkreslení) tím, že nechává narůst stromy hodně velké, zároveň udržuje snesitelnou varianci kombinováním výsledků jednotlivých stromů (většinové hlasování/průměrování). Oproti ostatním lesům je zde však snaha zajistit také nízkou korelaci

[10] Počet opakování má pro jednotlivé pozorování z L (pro $n \to \infty$) Poissonovo rozdělení se střední hodnotou 1. Pravděpodobnost, že pozorování nebude vybráno, je přibližně $e^{-1} \approx 0.37$.

mezi jednotlivými stromy. Pokud totiž tvoříme výběry s opakováním, které nejsou navzájem nezávislé, budou výsledné stromy korelované. Tato „podobnost“ jednotlivých stromů může vést k nadhodnoceným výsledkům klasifikace či predikce. Snížení korelace mezi stromy se dosáhne náhodným výběrem pouze určitého počtu prediktorů. Pro každý strom se tak nejlepší větvení pro daný uzel hledá pouze z m prediktorů $X_1,..,X_M$. Náhodný les tedy používá jak náhodný výběr pozorování, tak náhodný výběr prediktorů.

Algoritmus tvorby lesa můžeme popsat následovně:

1. Vytvoř bootstrapový podsoubor L_i o velikosti N - trénovací soubor.
2. Vyber náhodně m prediktorů.
3. Vytvoř strom T_i na bootstrapovém souboru L_i pouze s použitím m náhodně vybraných prediktorů (stejně jak bylo popsáno v metodě CART, hledáme nejlepší rozdělení daného uzlu mezi prediktory na dva dceřiné uzly). Růst stromu se zastaví, až strom dosáhne minimální hodnoty velikosti uzlu.
4. Zařaď *oob* pozorování (testovací soubor) vytvořeným stromem a urči výslednou klasifikační třídu (kategorii) nebo predikci všech *oob* pozorování.

Krok 1-4 se opakuje do konečného počtu stromů v lese.

5. Spočítej celkový výsledek klasifikace/predikce celého lesa většinovým hlasováním/průměrováním.

Pro algoritmus lesa je potřeba vybrat správný počet proměnných (m) pro náhodný výběr a počet stromů (*ntree*) v lese. Určení těchto parametrů je do jisté míry experimentální a vyžaduje zkušenosti. Klasickou cestou je provedení řady experimentů s různým nastavením těchto parametrů k získání lesa, který má nejmenší celkovou chybovost. Vzhledem k časově náročnému testování (zvláště jde-li o soubory obsahující tisíce záznamů) je vhodné vybrat takový počet stromů, který bude dostačující pro optimální klasifikaci. Na začátku tedy nastavíme počet stromů v lese na vyšší hodnotu (např. 20 násobek počtu prediktorů). Po určitém čase začínají stromy konvergovat ke správné hodnotě *oob* odhadu. Minimální velikost lesa lze určit jako počet stromů, kdy se chyba *oob* odhadu s přibývajícími stromy již nemění.

Dalším parametrem je počet náhodně vybraných prediktorů p.

Pro náhodné lesy je doporučeno následující nastavení:

- pro klasifikaci je hodnota $m = \sqrt{p}$ a minimální velikost uzlu je jedna;
- pro regresi je hodnota $m = p/3$ a minimální velikost koncového uzlu je pět.

Výše uvedené hodnoty slouží jako defaultní nastavení ve většině softwarů. V praxi však určení počtu prediktorů závisí na řešeném problému a parametr m je vhodný zvolit podle výsledků testování modelů s různým nastavením. Vybereme takové m, při němž má výsledný les nejmenší chybovost. Vzhledem k tomu, že stromy nelze přetrénovat, je počet prediktorů nejdůležitější hodnotou, kterou musíme zvolit, neboť počet stromů nás omezuje pouze časově.

5.1.1 Měření významnosti proměnných (*importance*)

Jak již bylo zmíněno, náhodné lesy (i další typy lesů) jsou vhodné pro úlohy, v nichž je velmi mnoho prediktorů. Ne všechny prediktory však nesou významnou informaci (v realitě

tomu bývá často naopak) a proto je pro interpretaci výsledků velmi užitečné zjistit, které proměnné jsou důležité. K tomu slouží měření významnosti proměnné.

Pomocí náhodných lesů můžeme spočítat hned několik měření významnosti. Nejčastěji se používá významnost založená na poklesu klasifikační přesnosti (*misclassification rate*), kdy jsou hodnoty prediktoru náhodně permutovány. Významnost lze také spočítat pomocí Gini indexu.

Významnost založená na randomizaci

Procedura měření významnosti založené na randomizaci se dá popsat následovně: po vytvoření *i*-tého stromu pro *i*-tý bootstrapový výběr jsou *oob* pozorování zařazena stromem do jednoho z terminálních uzlů a je určena přesnost jejich klasifikace/predikce (např. procento správně klasifikovaných pozorování). Následně jsou hodnoty *m*-tého prediktoru z *oob* výběru náhodně permutovány, a je opět pomocí příslušného stromu zjištěn výsledek klasifikace/predikce pro tato pozorování. Na konci jsou srovnány výsledky pozorování u *m*-tého prediktoru zatíženého šumem (randomizovaného) se správnou klasifikací těchto pozorování. Pokles v přesnosti predikce stromu, který nastane po randomizaci pozorování, je zprůměrován přes všechny stromy a je použit jako měření významnosti prediktorů. Tímto způsobem jsou zjištěny hodnoty *MR* (*misclassification rate*) pro každý prediktor, které určují jeho významnost. Toto měření je často vyjádřeno jako procento a je standardizováno na maximální hodnotu *MR* nejvýznamnějšího prediktoru (nejvýznamnější prediktor *MR* = 100%). Popsaný proces se opakuje pro všechny prediktory. Myšlenka celého postupu je jednoduchá: pokud „záměna“ hodnot proměnné nemá žádný vliv na výsledek, pak tato proměnná nemá význam. Čím větší je rozdíl mezi náhodou a skutečností, tím větší je i vliv proměnné. Randomizace má obdobný účinek jako nastavení koeficientů na nulu v lineární regresi (LR). Neměří se však efekt proměnné na predikci za její nepřítomnosti jako u LR. Pokud totiž odstraníme vybranou proměnnou při tvorbě stromu v náhodném lese, může být při predikci použita zástupná nebo kompetitivní proměnná. Měření významnosti nám však ukazuje predikční sílu dané proměnné. Protože je korelace hodnot *MR* mezi jednotlivými stromy poměrně nízká, lze spočítat rovněž směrodatnou odchylku (*SE*). Podělením hodnot *MR* pro každé pozorování jeho *SE* získáme *z*-skóre a za předpokladu normality můžeme zjistit statistickou významnost.

Pro každé pozorování můžeme navíc spočítat **lokální významnost** *m*-tého prediktoru. Je zjištěno procento správné klasifikace *n*-tého pozorování do správné kategorie přes všechny stromy, kdy bylo v *oob* výběru a hodnoty prediktoru *X* byly permutovány. Tato hodnota je odečtena od procenta správné klasifikace pro *oob* pozorování prediktoru bez randomizace.

Měření významnosti založené na randomizaci proměnných se objevuje v různých variantách.

Významnost založená na Gini idexu

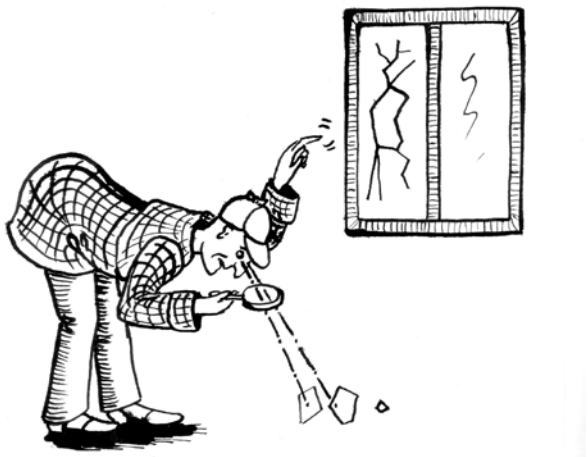
Při rozdělení uzlu na dva dceřiné uzly prediktorem, ke kterému je použit Gini index, dochází k poklesu tohoto indexu. Součet poklesu v *GI* (viz kriteriální statistika kap. 2.1) v jednotlivých stromech pro každý prediktor udává jeho významnost.

Pojďme se ještě podívat na případ, kdy hledáme optimální sadu prediktorů. Například u souboru pacientů chceme zjistit co nejmenší počet nutných vyšetření k určení diagnózy, aby nedocházelo ke zbytečné zátěži pacienta. Dostaneme však tuto informaci z významnosti proměnných? Pokud je proměnná významná, nemusí být nutně použita při tvorbě lesa. Stejně jako u stromů může jít o proměnnou zástupnou (s podobnou informací). Sada významných proměnných nemusí být počet proměnných použitých (či nutných) pro klasifikaci/predikci, je však sadou maximální. Nicméně většinou jsou všechny významné proměnné ve výsledném lese obsaženy, protože výběrem různých trénovacích souborů se nám liší i výsledné stromy. Pokud

jsou proměnné korelované a je mezi nimi jen malý rozdíl, budou se jejich role jako primární a zástupné proměnné střídat pro různé trénovací soubory. Ve výsledku však stačí použití jedné proměnné, neboť odstraněním její zástupné se přesnost výsledků výrazně nesníží. Ke zjištění nejmenší možné sady parametrů při zachování celkové přesnosti lesa je tedy nutné testovat různé kombinace proměnných. Můžeme tak rovněž dojít k více modelům o stejné přesnosti, ovšem s jinou kombinací proměnných.

V případě velkého počtu proměnných je les na začátku spuštěn jednou se všemi proměnnými a potom znovu s použitím pouze významných proměnných, čímž se výrazně šetří čas při testování optimálního lesa.

Příklad IX: Kriminalistika

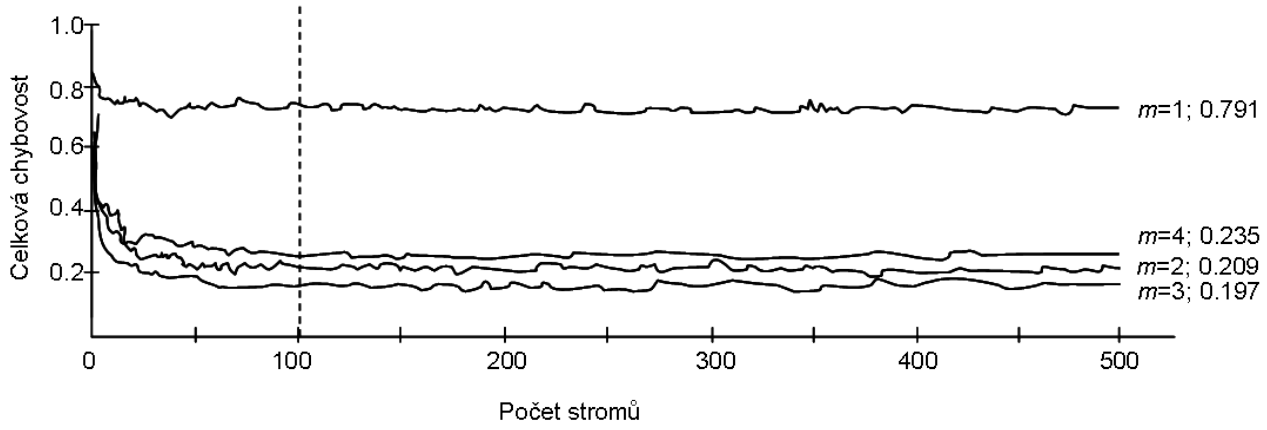


Studie různých typů klasifikací skla byla motivována vyšetřováním trestných činů. Sklo z místa činu může být použito jako důkaz, jestliže je ovšem správně identifikováno. Datový soubor obsahuje 214 vzorků šesti druhů různého skla a devět prediktorů – fyzikálně-chemické parametry skla: index lomu (RI), Na, Mg, Al, Si, K, Ca, Ba, Fe [20].

Závisle proměnná - typy skla

- A - okenní sklo (z domu)1
- B - okenní sklo (z domu)2
- C - okenní sklo (z auta)1
- D - obalové sklo
- E - sklo z nádobí
- F - reflektorové sklo

Nejprve otestujeme chybu lesa pro různé počty náhodně vybraných proměnných, nastavíme $m = \{1, 2, 3, 4\}$, a zároveň zjistíme počet stromů v lese dostačující pro klasifikaci (*ntree*) (obr. 5.1). Celková chybovost lesa pro klasifikaci skla byla stabilní již pro 100 stromů a dále se neměnila. Minimální nastavení hodnoty *ntree* je tedy 100. Protože však les nemůžeme přetrénovat a čas na výpočet pro takto malý datový soubor je zanedbatelný, použijeme hodnotu *ntree* = 500. Zvýšíme tak stabilitu významnosti proměnných, protože každá proměnná tak bude mít větší šanci být vybrána vícekrát a „prokázat“ svůj vliv. Celková chyba klasifikace byla nejnižší pro parametr $m = 3$. Nejhůře dopadla klasifikace pouze s jedním parametrem (79,1%), chybovost pro $m = 2$, 3 a 4 byla velmi podobná a pro větší počet proměnných by dále zvolna stoupala.



Obr. 5.1 *Výsledek klasifikace pro různé hodnoty parametru m.*

Níže je výsledek z náhodného lesa s použitím 500 stromů a $m = 3$.
Podívejme se na klasifikační chybu pro jednotlivé skupiny (tab. 5.1).

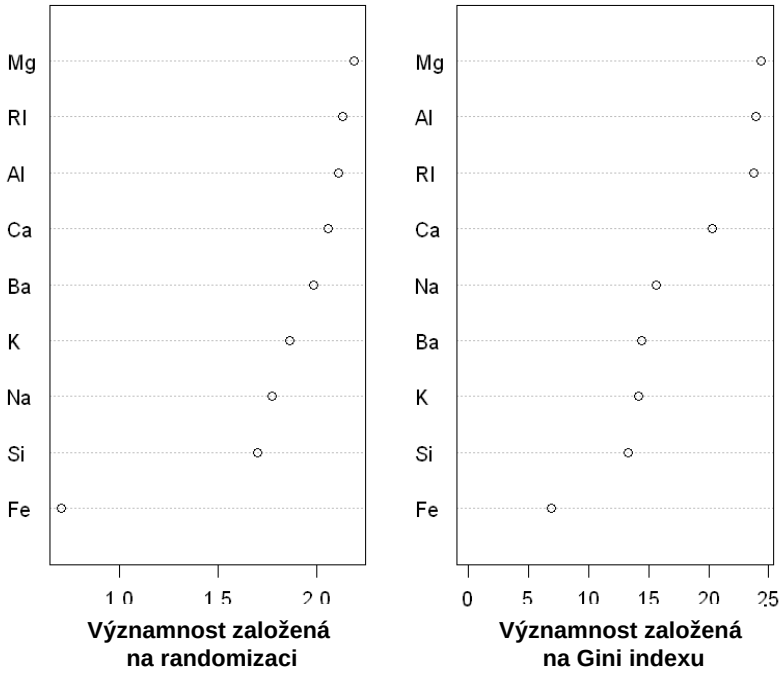
Tabulka 5.1 *Procento klasifikace pro jednotlivé typy skla.*

Přesnost klasifikace (%)	Počet vzorků	Typ skla	A	B	C	D	E	F
64.3	70	**A**	45	7	16	0	1	1
59.2	76	**B**	8	45	9	7	6	1
76.5	17	**C**	4	0	13	0	0	0
92.3	13	**D**	0	0	0	12	0	1
100	9	**E**	0	0	0	0	9	0
89.7	29	**F**	1	0	0	2	0	26

Výsledky ukazují, že nejhůře se dá identifikovat okenní sklo z domu (A a B) 64,3 % a 59,2% , které je podle fyzikálně-chemických vlastností podobné okennímu sklu z auta (C), kam jsou vzorky často chybně klasifikovány. I v samotném typu okenního skla z aut nastává překryv. Žádné z typů skla není jednoznačně vymezeno. Přestože jsme schopni zařadit správně všechny vzorky ze skla z nádobí (E), stále mohou být neznámé vzorky identifikovány (i když s malou pravděpodobností) jako okenní sklo z domu (A i B). Z toho pohledu je nejlépe definována kategorie F (reflektorové sklo), která má vysoké procento správné klasifikace a zároveň se do ní příliš nezařazují vzorky z jiných kategorií.

Přesnost klasifikace celého lesa lze jednoduše spočítat jako procento správně klasifikovaných vzorků. V našem případě bylo správně klasifikováno 177 vzorků skla a přesnost klasifikace je 82,7%. Tento postup je však vhodný pouze v případě, kdy je počet vzorků ve všech kategoriích stejný. Při nestejné velikosti kategorií musíme dát všem kategoriím stejnou váhu, spočítáme tedy procento klasifikace pro každou kategorii zvlášť a výsledkem bude průměr z klasifikací jednotlivých kategorií. V našem případě přibližně 80,3%.

Nyní bude zajímavé zjistit, které proměnné jsou pro klasifikaci důležité (obr. 5.2).



Obr. 5.2 *Významnost fyzikálně chemických parametrů skla pro klasifikaci. Proměnné jsou seřazeny sestupně podle významnosti.*

Významnost proměnných založených na randomizaci a Gini indexu je velmi podobná. Mezi nejvýznamnější proměnné patří mangan, hliník a index lomu. Naopak nejméně významnou proměnnou pro klasifikaci je železo.

5.1.2 Efekt proměnných na predikci

Při použití lesa pro klasifikaci můžeme o prediktorech získat další užitečné informace. Mimo významnosti proměnných by nás mohlo zajímat, pro kterou kategorii nebo kategorie je významná. Pro případy z *oob* výběru známe kategorii, do které bylo pozorování zařazeno, můžeme zjistit podíl klasifikace pozorování do jednotlivých kategorií neboli *cpv* (*class proportion vote*). Uveďme jednoduchý příklad pro čtyři kategorie A, B, C, D kdy by bylo pozorování při klasifikaci ze 100 stromů zařazeno 10 stromy do A, 50 do B a 40 do C. Hodnoty *cpv* pro jednotlivé kategorie budou: cpv_A=0,1, cpv_B = 0,5, cpv_C = 0,4 a cpv_D = 0.

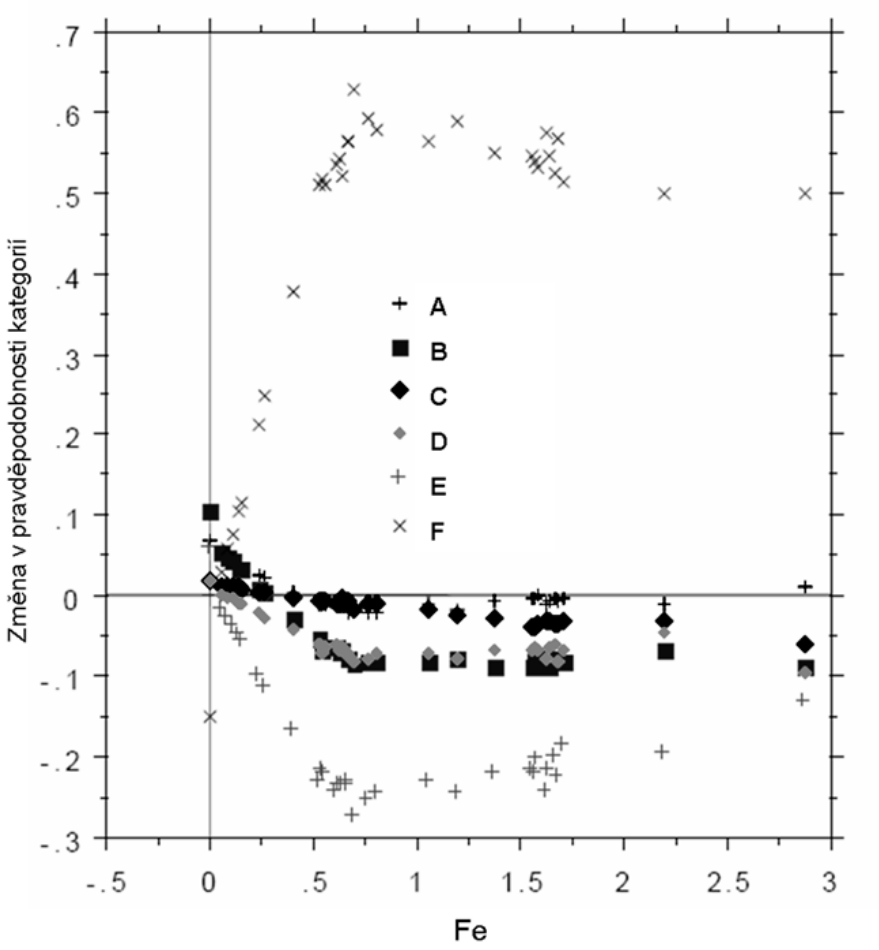
Pro každou kategorii *j* a každou proměnnou *m* spočítáme pravděpodobnost zařazení každého vzorku do kategorie *j*, kdy je *m*-tá proměnná *X* randomizována. Rozdíl mezi cpv_{Ran} pro randomizovanou proměnnou a proměnnou bez šumu udává velikost změny pravděpodobnosti zp_{cpv} pro každou kategorii u všech vzorků:

$$zp_{cpv} = cpv - cpv_{Ran} \tag{5.3}$$

Myšlenka je tedy stejná jako při určení významnosti proměnné.

Vyneseme-li hodnoty této změny do grafu proti hodnotám proměnné, získáme graf efektu proměnné na predikci.

Na obrázku 5.3 je zobrazena nejméně významná proměnná (Fe) z našeho kriminalistického případu.



Obr. 5.3 *Pokles v pravděpodobnosti kategorií po randomizaci proměnné Fe [21].*

Železo je významné pouze pro odlišení reflektorového skla. Pokud jsou však hodnoty železa velmi nízké, nejsme schopni rozlišit ani tuto kategorii, železo tedy musí být přítomno ve vyšších hodnotách. Mohli bychom se podívat, zda jsme reflektorové sklo schopni odlišit pomocí jiné proměnné. V takovém případě by železo nebylo nutné při identifikaci různých typů skel. Záporné hodnoty indikují, že by klasifikace při odstranění této proměnné byla pro ostatní třídy přesnější. Dodejme ještě, že záporné hodnoty mohou nastat pouze tehdy, pokud je pravděpodobnost vzorku pro danou kategorii u randomizované proměnné vyšší, než při správném pořadí hodnot této proměnné.

5.1.3 Těsnost (*proximity*)

Jak již bylo zmíněno v úvodu kapitoly, náhodné lesy je možné využít rovněž jako techniku pro ordinační či shlukové analýzy, k čemuž slouží měření těsnosti. Protože stromy se nechávají růst velké (obvykle je počet pozorování v koncovém uzlu 5 nebo dokonce 2) a nejsou zpětně prořezávány, je zřejmé, že si pozorování ve stejném terminálním uzlu budou velmi podobná. Po vytvoření všech stromů v lese jsou pozorování (trénovací i testovací) zařazena každým stromem. Pokaždé, když se pozorování ocitnou ve stejném terminálním uzlu, vzroste jejich těsnost o 1. Pro každý pár pozorování je tedy možné spočítat, kolikrát se vyskytl ve stejném terminálním uzlu přes všechny stromy. Na konci se tyto hodnoty normalizují poděleným celkovým počtem stromů.

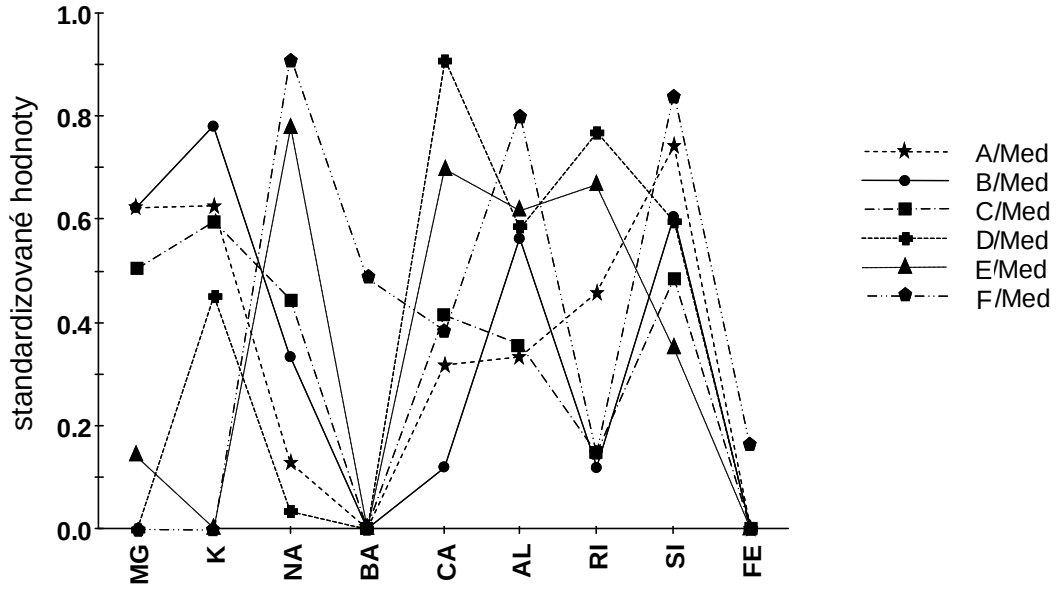
Např. jestliže se pozorování vyskytla spolu v terminálním uzlu 80 x ze 100 stromů, je těsnost rovna 0,8. Hodnoty těsnosti jsou v rozmezí 0 - 1. Přičemž 1 značí maximální těsnot (ve všech stromech byla pozorování vždy zařazena do stejného uzlu) a 0 minimální (pozorování se nikdy nevyskytla ve stejném uzlu). Měření těsnosti mezi pozorováními *n* a *k* tak tvoří matici těsností $\boldsymbol{M}_{prox} = \{prox(n, k)\}$. Velikost matice je *N*x*N*. Takto vytvořená matice je symetrická, pozitivně definitní a nabývá hodnot mezi 0-1 s prvky na diagonále rovnými jedné. Tuto matici podobnosti ($\boldsymbol{M}_{prox,}$) popřípadě vzdálenosti $(1-\boldsymbol{M}_{prox})$ je možné použít v různých shlukovacích a ordinačních metodách. Nejčastější využití je pro výpočet faktorových os ve vícerozměrném škálování, které umožňuje projekci vzorků do prostoru s méně dimenzemi, přičemž zachovává vzdálenosti mezi objekty. Tento postup může být velmi užitečný pro vizualizaci výsledků klasifikace nebo k hodnocení překryvu jednotlivých skupin [22][23].

5.1.4 Prototypy kategorií

Dalším příkladem použití měření těsnosti jsou prototypy kategorií. Pro každou kategorii *j* najdeme pozorování, u něhož je největší počet pozorování z téže kategorie mezi jeho *k* nejbližšími sousedy, které jsou definovány na základě měření těsnosti. Mezi těmito *k*-pozorováními zjistíme medián a kvartily pro každý prediktor. Mediány jsou prototypy dané kategorie a kvartily nám dávají odhad jejich stability. Hodnoty jsou standardizovány (odečtením 5. percentilu a podělením rozsahem mezi 5. a 95. percentilem). Pro kategoriální proměnné je prototypem kategorie s největší frekvencí. Druhý prototyp můžeme získat zopakováním této procedury, ale již bez pozorování, která byla obsažena v prvním prototypu.

K určení prototypů můžeme místo měření těsnosti použít také pravděpodobnost zařazení do správné kategorie. Zvolíme si hodnotu této pravděpodobnosti například větší než 0,5 (pozorování bylo s více než 50% zařazeno správně) nebo 1 (pozorování bylo všemi stromy klasifikováno do správné kategorie). Z pozorování, která mají pravděpodobnost zařazení do správné kategorie větší než zadaná hodnota, můžeme opět spočítat medián a kvartily (v případě spojitého prediktoru) nebo procentuální zastoupení (pro kategoriální prediktor). Prototypy tak udávají celkovou představu o tom, jaký vztah mají prediktory ke klasifikaci a tvoří „jádro“ dané kategorie.

Hodnoty prototypů fyzikálně-chemických parametrů skla z příkladu IX jsou zobrazeny na obrázku 5.4.



Obr. 5.4 *Prototypy parametrů různých druhů skla. Například nejvýznamnější proměnná mangan rozděluje kategorie na dvě skupiny. Vyšších hodnot nabývá pro kategorie A,B,C a nižších hodnot pro D,E,F.*

5.1.5 Překryv kategorií

K určení překryvu jednotlivých kategorií můžeme použít zobrazení mapy těsnosti (*proximity heat map*). K tomu využijeme matici těsnosti. Intenzita barev v mapě je dána vzrůstající mírou těsností jednotlivých vzorků. Stejně lze použít k vytvoření mapy i matici pravděpodobností (*probality heat map*).

Následující příklad je ukázkou použití náhodného lesa pro malý datový soubor a větší počet prediktorů.

Příklad X: Výběr indikačních taxonů makrozoobentosu pro říční habitaty

Kategoriální závisle proměnná obsahuje 58 pozorování rozdělených do 4 typů habitatů definovaných na základě polohy odběrových míst v rámci koryta toku a lokálních hydraulických podmínek. Jednalo se o peřeje (CH_RNRF), příbřežní tišiny (M_POOL), tůně (CH_POOL) a boční ramena (S_POOL). Prediktory jsou abundance 87 taxonů makrozoobentosu, zjištěných v těchto vzorcích. Cílem studie bylo stanovit preference taxonů k abioticky definovaným říčním habitatům a schopnost taxonů odlišit mezi jednotlivými typy habitatů [24].

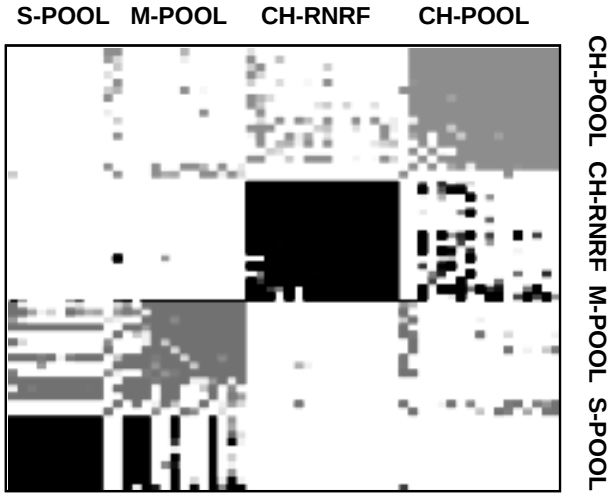
Z výsledků Random Forest bylo zjištěno procento správně zařazených pozorování *OA* pro jednotlivé habitaty (tab. 5.2).

Tabulka 5.2 *Procento klasifikace pro různé habitaty.*

Typ habitatu	Počet pozorování	OA (%)	CH-POOL	CH-RNRF	M-POOL	S-POOL
CH-POOL	17	53	9	6	2	0
CH-RNRF	16	87	2	14	0	0
M-POOL	15	54	2	0	8	5
S-POOL	10	90	0	0	1	9
Celkem	58	69	13	20	11	14

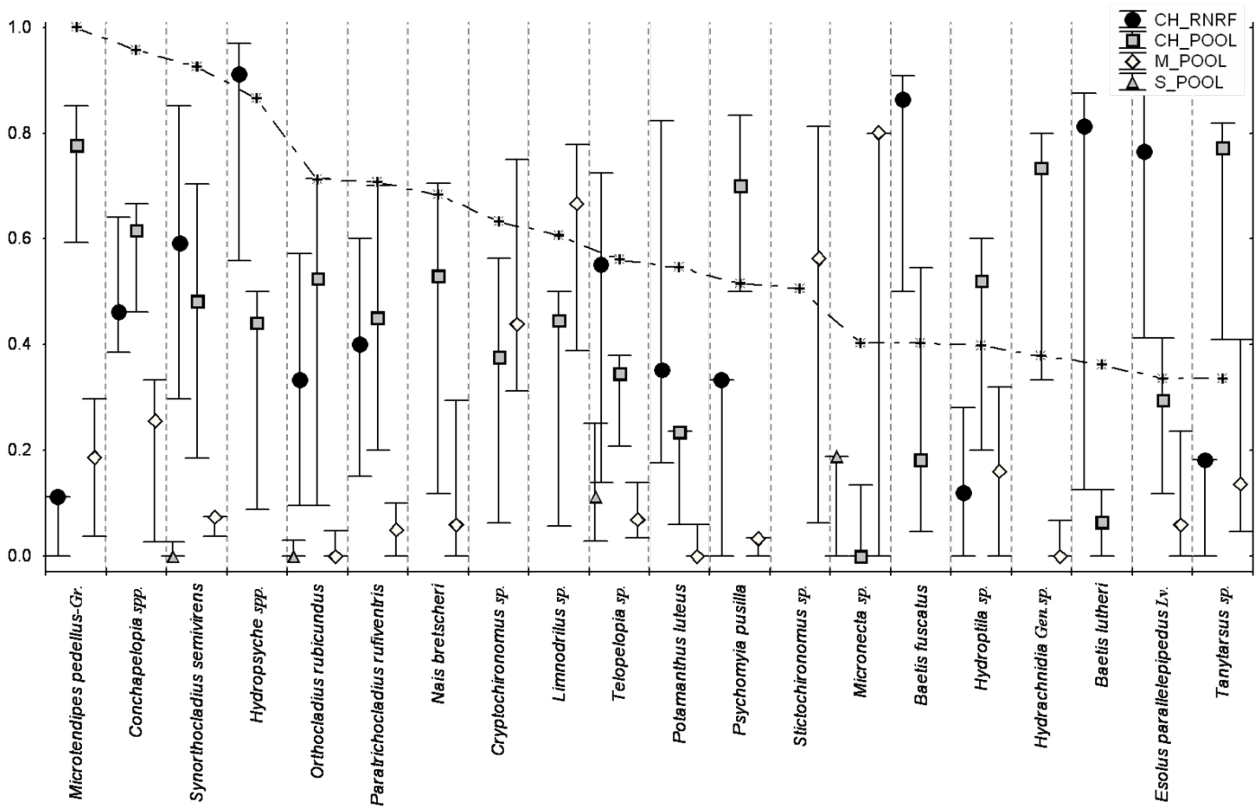
Nejlépe definované habitaty z hlediska taxonů byly peřeje (*OA* = 87%) a boční ramena (*OA* = 90%). Naopak tůně (*OA* = 53%) a příbřežní tišiny (*OA* = 54%) byly odlišitelné hůře, vzhledem k jejich překryvu s ostatními habitaty.

Překryv habitatů je dobře viditelný z PHM (*proximity heat map*) založené na matici těsnosti (obr. 5.5).



Obr. 5.5 *PHM znázorňuje klasifikaci a překryv pozorování v habitatech. Pozorování v matici těsnosti jsou seřazena podle příslušnosti k daným habitatům tak, že nejvíce podobná pozorování leží na diagonále. Pozorováním jsou přiřezena odstíny šedi podle hodnoty těsnosti (čím intenzivnější odstín, tím vyšší hodnota).*

K určení významnosti taxonů bylo použito měření významnosti založené na randomizaci. Pomocí klasifikačního lesa bylo vybráno 50 významných indikačních taxonů. Pro každý taxon byla zjištěna míra preference ke každému typu habitatu pomocí prototypů (obr. 5.6).



Obr. 5.6 *Prototypy taxonů (zobrazeno pro prvních 20 taxonů) – je zobrazen medián a kvartily logaritmicky transformovaných abundancí. Pořadí taxonů je dáno jejich významností. Prototypy byly spočítány pro každý habitatový typ: peřeje (kolečka), tůně (čtverečky), příbřežní tišiny (kosočtverečky) a boční ramena (trojúhelníčky).*

Z výsledků vyplynulo, že bylo velmi málo taxonů preferujících příbřežní tišiny a boční ramena. Tyto habitaty byly definovány převážně „negativně“ indikačními taxony (správná klasifikace vzorků byla způsobena nevýskytem nebo nízkou abundancí taxonů). Ukázalo se, že většina taxonů nebyla specifická pro jeden habitat, ale preferovaly dva nebo tři habitaty (tab. 5.3).

Tabulka 5.3 *Hodnoty preferencí taxonů pro jednotlivé habitaty jsou mediány abundancí z nejvíce pravděpodobných pozorování pro daný habitat (prototypy). Max. habitat – habitat s nejvyšší hodnotou abundance; uvedená významnost je založená na randomizaci hodnot abundancí taxonů.*

Taxon	Počet výskytů taxonu	Významnost (%)	Preference taxonů				Max. habitat
			CH_RNRF	CH_POOL	M_POOL	S_POOL	
Microtendipes pedellus-Gr.	47	100,00	0,11	0,78	0,19	0,00	CH_POOL
Conchapelopia spp.	52	95,82	0,46	0,62	0,26	0,00	CH_POOL
Synorthocladius semivirens	44	92,65	0,59	0,48	0,07	0,00	CH_RNRF
Hydropsyche spp.	64	86,47	0,91	0,44	0,00	0,00	CH_RNRF
Orthocladius rubicundus	41	71,26	0,33	0,52	0,00	0,00	CH_POOL
Paratrichocladius rufiventris	42	70,65	0,40	0,45	0,05	0,00	CH_POOL
Nais bretscheri	29	68,35	0,00	0,53	0,06	0,00	CH_POOL
Cryptochironomus sp.	33	63,41	0,00	0,38	0,44	0,00	M_POOL
Limnodrilus sp.	53	60,61	0,00	0,44	0,67	0,11	M_POOL
Telopelopia sp.	48	56,01	0,55	0,34	0,07	0,00	CH_RNRF
Potamanthus luteus	36	54,68	0,35	0,24	0,00	0,00	CH_RNRF
Psychomyia pusilla	38	51,49	0,33	0,70	0,03	0,00	CH_POOL
Stictochironomus sp.	22	50,57	0,00	0,00	0,56	0,19	M_POOL
Micronecta sp.	28	40,36	0,00	0,00	0,80	0,00	M_POOL
Baetis fuscatus	32	40,27	0,86	0,18	0,00	0,00	CH_RNRF
Hydroptila sp.	39	39,75	0,12	0,52	0,16	0,00	CH_POOL
Hydrachnidia Gen.sp.	32	37,90	0,00	0,73	0,00	0,00	CH_POOL
Baetis lutheri	26	36,21	0,81	0,06	0,00	0,00	CH_RNRF
Esolus parallelepipedus Lv.	37	33,53	0,76	0,29	0,06	0,00	CH_RNRF
Tanytarsus sp.	51	33,46	0,18	0,77	0,14	0,05	CH_POOL

5.1.6 Detekce odlehlých hodnot

Pomocí měření těsnosti můžeme rovněž definovat odlehlá pozorování. Za odlehlé pozorování je považováno takové, které má nejmenší těsnost k ostatním pozorováním ve své klasifikační kategorii, je tedy definováno pouze pro kategorii, do které patří.

Definujme průměrnou těsnost pozorování *n* od všech pozorování *k* ve stejné kategorii *j* jako:

$$\overline{P}(n) = \sum_{c(k)=j} prox^2(n,k) \tag{5.4}$$

Míra odlehlosti pro pozorování *n* je definována jako:

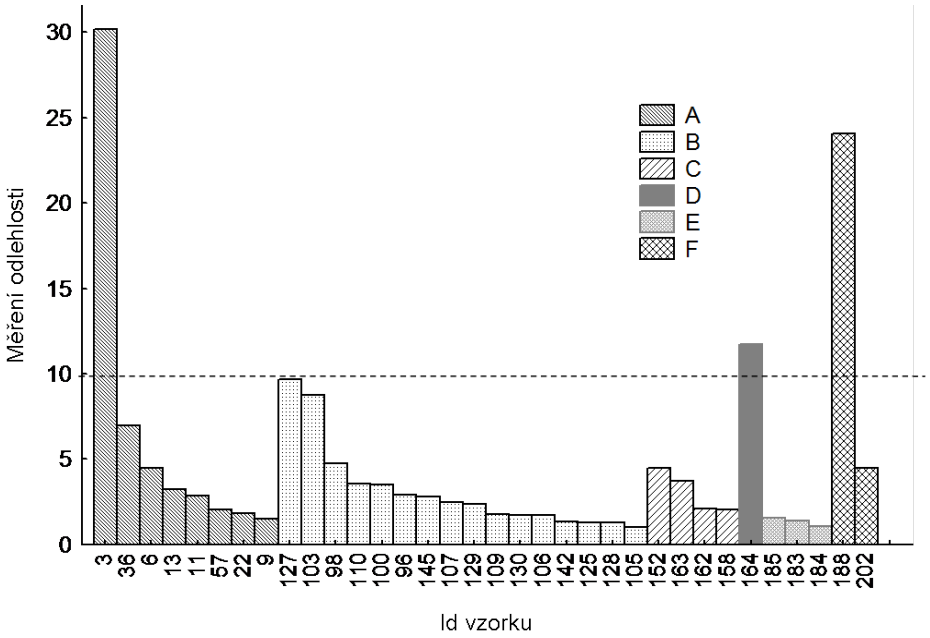
$$out(n) = N_j \big/ \overline{P}(n), \tag{5.5}$$

kde N_j je počet pozorování v kategorii *j*. Hodnota *out(n)* bude velká, jestliže průměrná těsnost *n* k dalším pozorováním *k* ve stejné kategorii bude malá. Pro všechna pozorování v dané kategorii

spočítáme medián ze všech $out(n_j)$ a jejich absolutní odchylku od mediánu. Odečtením mediánu od hodnoty $out(n)$ a podělením její absolutní odchylkou získáme finální měření odlehlosti pozorování n.

Hodnoty $out(n) < 0$ jsou převedeny na nulu. Pokud je $out(n) > 10$, pozorování je považováno za odlehlé.

Vrátíme-li se zpátky k našemu kriminalistickému příkladu IX, zjistíme, že pouze tři měření jsou podle pravidla $out(n) > 10$ odlehlé a to v kategorii A, E a F (obr. 5.7). Větší množství vzorků, které mají malou těsnost k ostatním vzorkům v kategorii, je přítomno ve skupině A a B, což jsou kategorie, které byly nejhůře klasifikovány. V těchto kategoriích je velký rozptyl hodnot v jejich prototypech.



Obr. 5.7 *Detekce odlehlých hodnot pro jednotlivé typy skla.*

5.1.7 Chybějící hodnoty

Použitím náhodných lesů pro klasifikaci můžeme doplnit chybějící hodnoty. Jsou dvě možnosti, jak les tuto náhradu provádí. První, jednodušší, rychlejší, ale méně přesná cesta je nahrazení chybějící hodnoty x_n mediánem hodnot m-té proměnné v kategorii j závisle proměnné. Pokud je proměnná kategoriální, je k doplnění použita hodnota kategorie s nejvyšší frekvencí opět pouze v příslušné kategorii závisle proměnné.
Druhou možností je využití měření těsnosti. Tato varianta je poměrně přesná a vhodná pro datové soubory, které obsahují velké množství chybějících hodnot, nicméně může značně zvýšit výpočetní čas, protože jde o iterativní proces. Chybějící hodnota x_i m-té proměnné X je nahrazena váženým průměrem pozorování x_k, jejichž hodnoty byly vyplněné. Jako váha je použito měření těsnosti $prox(x_i,x_k)$. U kategoriální proměnné je chybějící hodnota nahrazena nejvíce frekventovanou hodnotou, která je opět vážená těsností. Nahrazené hodnoty jsou použity v další iteraci lesa a jsou spočítány nové těsnosti. Tento proces se zastaví, pokud již nedochází k žádnému zlepšení, nebo např. po pěti iteracích (počet iterací je rovněž možné zvolit).

Příklad XI: Random Forest - Mořští krabi

Ukázkový příklad ke cvičení v programu R.

Datový soubor obsahuje 200 měření a 8 prediktorů popisujících 5 morfologických měření 200 krabů (*Leptograpsus variegatus*) rozdělených do dvou barevných variant. Každá varianta je rovnoměrně zastoupena oběma pohlavími. Zajímá nás, zda je možné kraby pomocí morfologických měření rozlišit. Měření proběhlo na krabech nasbíraných v přístavním městě Fremantle v západní Austrálii [25] [26].

Popis prediktorů:
sp - druh s kategoriemi *B* nebo *O* pro modrou a oranžovou variantu
sex - pohlaví kódované jako *M* a *F*
index - index 1:50 - id pozorování uvnitř čtyřech skupin *sp* x *sex*
FL - frontal lobe size (mm) - velikost čelního laloku
RW- rear width (mm) - šířka zadní části (zadečku)
CL - carapace length (mm) - délka krunýře
CW - carapace width (mm) - šířka krunýře
BD - body depth (mm) - výška trupu

Nejdříve načteme knihovnu MASS, která obsahuje výše popsaný datový soubor se jménem *crabs*:

```
> library(MASS)
> data(crabs)
```

Zobrazíme si základní popisnou statistiku souboru:

```
> crabs

    sp sex index   FL   RW   CL   CW   BD
1    B   M     1  8.1  6.7 16.1 19.0  7.0
2    B   M     2  8.8  7.7 18.1 20.8  7.4
3    B   M     3  9.2  7.8 19.0 22.4  7.7
4    B   M     4  9.6  7.9 20.1 23.1  8.2
5    B   M     5  9.8  8.0 20.3 23.0  8.2
6    B   M     6 10.8  9.0 23.0 26.5  9.8
7    B   M     7 11.1  9.9 23.8 27.1  9.8
8    B   M     8 11.6  9.1 24.5 28.4 10.4
9    B   M     9 11.8  9.6 24.2 27.8  9.7
10   B   M    10 11.8 10.5 25.2 29.3 10.3
…
```

```
> summary(crabs)

sp      sex         index            FL              RW              CL
 B:100   F:100   Min.   : 1.0   Min.   : 7.20   Min.   : 6.50   Min.   :14.70
 O:100   M:100   1st Qu.:13.0   1st Qu.:12.90   1st Qu.:11.00   1st Qu.:27.27
                 Median :25.5   Median :15.55   Median :12.80   Median :32.10
                 Mean   :25.5   Mean   :15.58   Mean   :12.74   Mean   :32.11
                 3rd Qu.:38.0   3rd Qu.:18.05   3rd Qu.:14.30   3rd Qu.:37.23
                 Max.   :50.0   Max.   :23.10   Max.   :20.20   Max.   :47.60
```

```
CW              BD
 Min.   :17.10   Min.   : 6.10
 1st Qu.:31.50   1st Qu.:11.40
 Median :36.80   Median :13.90
 Mean   :36.41   Mean   :14.03
 3rd Qu.:42.00   3rd Qu.:16.60
 Max.   :54.60   Max.   :21.60
```

Načteme knihovnu pro tvorbu náhodného lesa *randomForest*:

```
> library(randomForest)
```

V náhodném lese je možno nastavit hodnoty parametrů, které specifikují výsledný les a jeho výstupy. Defaultní nastavení hodnot je následující:

`randomForest(x, y=NULL,`
`xtest=NULL, ytest=NULL,` / testovací soubor není předem zadán;
`ntree=500,` / počet stromů v lese;
`mtry=if (!is.null(y) && !is.factor(y))max(floor(ncol(x)/3), 1) else floor(sqrt(ncol(x))),` /počet náhodně vybraných proměnných; defaultní hodnoty jsou $\sqrt{p}$ pro klasifikaci a $p/3$ pro regresi, kde p je počet prediktorů;
`replace=TRUE` /pozorování může být vybráno vícekrát (při procesu rozdělení na *oob* vzorky);
`classwt=NULL` /váha jednotlivých kategorií závisle proměnné, defaultně mají všechny stejnou váhu;
`strata,` `sampsize = if (replace) nrow(x) else ceiling(.632*nrow(x)),` /parametry pro stratifikovaný výběr;
`nodesize = if (!is.null(y) && !is.factor(y)) 5 else 1,` /minimální počet vzorků v terminálním uzlu, defaultně 1 pro klasifikaci a 5 pro regresi;
`maxnodes = NULL,` /maximální počet terminálních uzlů stromu;
`importance=FALSE,` /výpočet významnosti proměnných;
`localImp=FALSE,` /významnost každého pozorování;
`nPerm=1,` /počet iterací, kdy jsou *oob* pozorování permutovány pro výpočet *importance* proměnných, zatím pouze pro regresi;
`proximity,` /výpočet matice těsnosti;
`oob.prox=proximity,` /matice těsnosti pouze pro *oob* pozorování;
`norm.votes=TRUE,` /výsledné hlasování je vyjádřeno jako podíl, jinak přímo počet;
`do.trace=FALSE,` /zobrazí výstupy procesu hledání;
`keep.forest=!is.null(y) && is.null(xtest), corr.bias=FALSE, keep.inbag=FALSE, ...)` /FALSE – výsledek lesa nebude uložen ve finálním výstupu.

Nyní vytvoříme vlastní náhodný les. Začneme rozlišením dvou barevných variant krabů bez ohledu na pohlaví:

```
> set.seed(12)
> les_krabi <- randomForest(sp ~ FL+RW+CL+CW+BD, data=crabs,
importance=TRUE, proximity=TRUE)
```

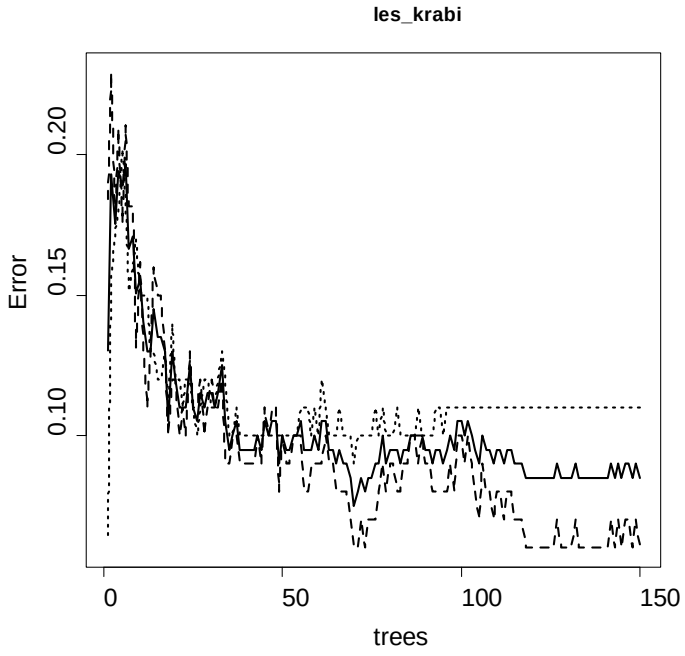
Před každým spuštěním lesa je vhodné zadat jiné náhodné číslo *set.seed*, aby byla zajištěna větší „náhodnost“ algoritmu. V lese potřebujeme vybrat náhodně *oob* vzorky i počet prediktorů a algoritmus musí odněkud začít.

Do proměnné *les_krabi* uložíme výsledek náhodného lesa, který spustíme pomocí funkce *randomForest*, specifikujeme závisle proměnnou *sp* a prediktory z datového souboru *crabs*, zadáme výpočet významnosti (*importance*) proměnných a matici těsnosti (*proximity*). Zobrazíme výsledek výpočtu:

```
> print(les_krabi)

Call:
 randomForest(formula = sp ~ FL + RW + CL + CW + BD, data = crabs,
importance = TRUE, proximity = TRUE)
                Type of random forest: classification
                      Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 10.5%
Confusion matrix:
   B  O class.error
B 90 10         0.11
O 10 90         0.11
```

Les obsahuje 500 stromů a počet proměnných 2, celková chyba lesa je 11%, krabi B i O mají stejné procento správně zaklasifikovaných vzorků 89%. Podívejme se nyní na závislost počtů stromů v lese na celkové chybě lesa. Zobrazíme výsledek pro prvních 150 stromů:

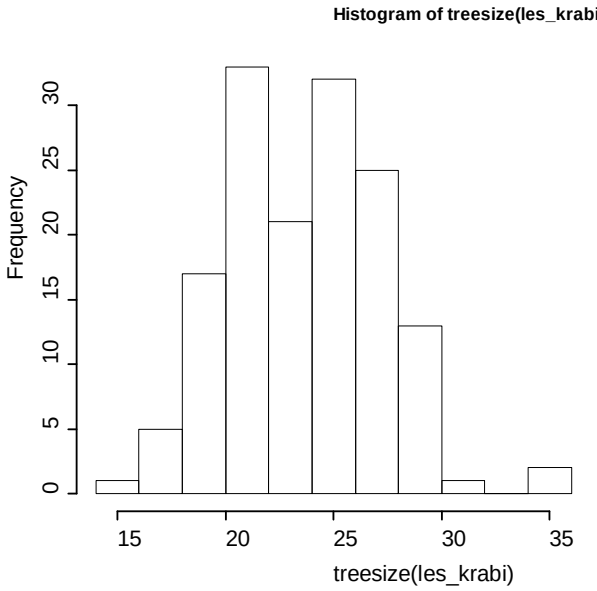
```
> set.seed(52)
> les_krabi <- randomForest(sp ~ FL+RW+CL+CW+BD, data=crabs, ntree=150)
> plot(les_krabi, cex.axis=1.5, cex.lab=1.5, lwd=2, col='black')
```



Výsledek klasifikační chyby je stabilní již pro poměrně nízký počet stromů (přibližně od 70 stromů). Dostačují hodnota *ntree* = 100.

Další zajímavou informací je velikost výsledných stromů, kterou můžeme orientačně zjistit z histogramu počtu terminálních uzlů. Průměrná velikost stromů se pohybuje kolem 25 uzlů.

```
> hist(treesize(les_krabi))
```

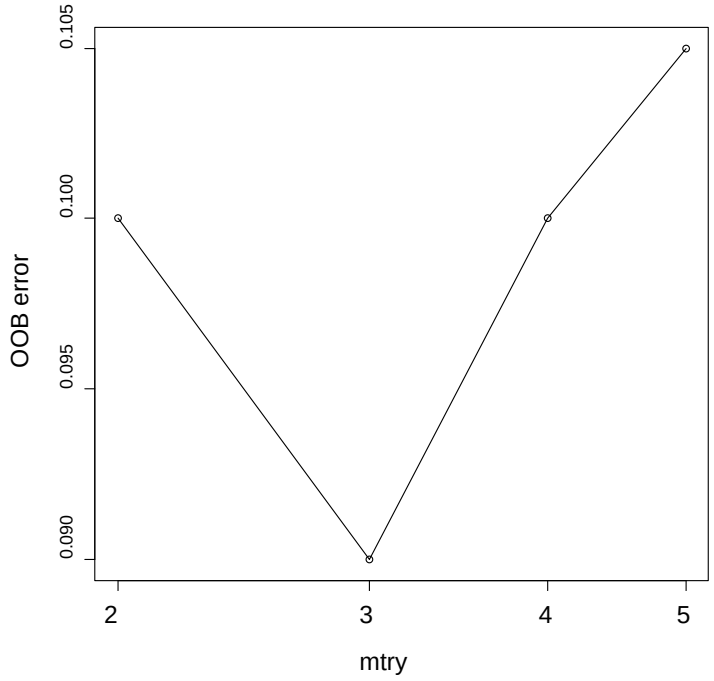


Pomocí funkce *getTree* si můžeme zobrazit i vybraný strom. V tomto případě vybereme dvacátý. Výstup bude v textové podobě. Vzhledem k velké velikosti stromů však bývá často nepřehledný.

```
> getTree(randomForest(sp ~ FL+RW+CL+CW+BD, data=crabs, ntree=100), 20,
labelVar=TRUE)
```

Zbývá nastavit nejdůležitější proměnnou pro vytvoření lesa - počet náhodně vybraných proměnných (prediktorů), na základě kterých se budou jednotlivé stromy dělit. Zde je tento parametr označen jako *mtry*. Defaultní hodnoty parametru *mtry* jsou $\sqrt{p}$ pro klasifikaci a $p/3$ pro regresi, kde p je počet prediktorů. Nastavitelným parametrem je *improve*. Aby probíhalo další vyhledávání optimálního počtu prediktorů, musí být zlepšení *oob* chyby větší než jeho stanovená hodnota. Parametr *trace* zobrazí výstupy procesu hledání. Poněkud záludný je parametr *stepFactor*, který určuje, o kolik se bude parametr *mtry* zvyšovat (nebo snižovat) při testování jeho vlivu na výslednou klasifikaci.

```
> set.seed(126)
> mtryles <- tuneRF(crabs[,4:8], crabs[,1], stepFactor=1.5, improve=0.05,
ntree=100, mtry=4, trace=TRUE, plot=TRUE)

mtry = 4  OOB error = 10%
Searching left ...
mtry = 3        OOB error = 9%
0.1 0.05
mtry = 2        OOB error = 10%
-0.1111111 0.05
Searching right ...
mtry = 5        OOB error = 10.5%
-0.1666667 0.05
```



Rozdíly v použití různé hodnoty *mtry* jsou velmi malé. Pokud začneme vždy od různého náhodného čísla, výsledky se mohou lišit i při stejném nastavení parametrů funkce. Při menší hodnotě *mtry* jsou stromy méně korelované, můžeme tedy zvolit mezi hodnotou 2 a 3.

Nyní již můžeme spustit les s optimálním nastavením hodnot parametrů:

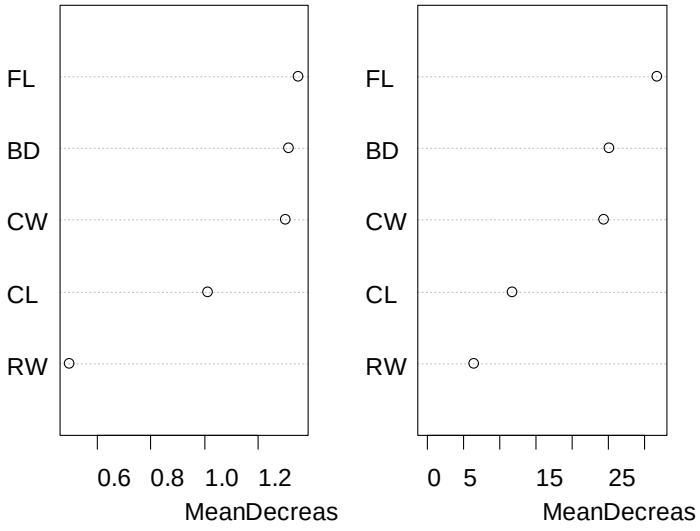
```
> set.seed(10)
> randomForest(formula = sp ~ FL + RW + CL + CW + BD, data = crabs,
importance = TRUE, proximity = TRUE, ntree = 100, mtry =3)

Call:
 randomForest(formula = sp ~ FL + RW + CL + CW + BD, data = crabs,
importance = TRUE, proximity = TRUE, ntree = 100, mtry = 3)
               Type of random forest: classification
                     Number of trees: 100
No. of variables tried at each split: 3

        OOB estimate of  error rate: 10.5%
Confusion matrix:
   B  O class.error
B 89 11        0.11
O 10 90        0.10
```

V tomto případě se výsledek klasifikace téměř nezměnil od původního nastavení hodnot. Procento správně klasifikovaných *oob* vzorků při velikosti lesa 100 stromů a třech náhodně vybraných prediktorech je 89.5%.

Poté, co jsme nastavili optimální hodnoty pro tvorbu lesa, můžeme použít další funkce, které jsou důležité zejména pro interpretaci našich výsledků. Víme, že dvě varianty krabů jsme schopni klasifikovat s vysokou přesností. Bylo by zajímavé se podívat, které proměnné jsou pro determinaci nejdůležitější. Hodnoty významnosti proměnných zjistíme pomocí funkce *importance* a v grafické podobě je zobrazíme funkcí *varImpPlot*. U funkce *importance* můžeme

zvolit významnost založenou na Gini indexu (*type* = 2) nebo na poklesu klasifikační chyby při randomizaci proměnné (*type* = 1).

```
> importance(les_krabi, type=2)
> varImpPlot(les_krabi)

MeanDecreaseGini                    MeanDecreaseAccuracy
FL        27.551568            FL             2.408711
RW         9.932005            RW             1.360970
CL        15.852326            CL             1.963287
CW        22.109231            CW             2.283460
BD        24.036957            BD             2.363172
```



Nejdůležitějšími proměnnými pro odlišení modré a oranžové varianty kraba jsou výška čelního laloku (FL), šířka trupu (BD) a šířka krunýře (CW). Zobrazíme, kolikrát byly jednotlivé proměnné použité ve stromech při tvorbě lesa:

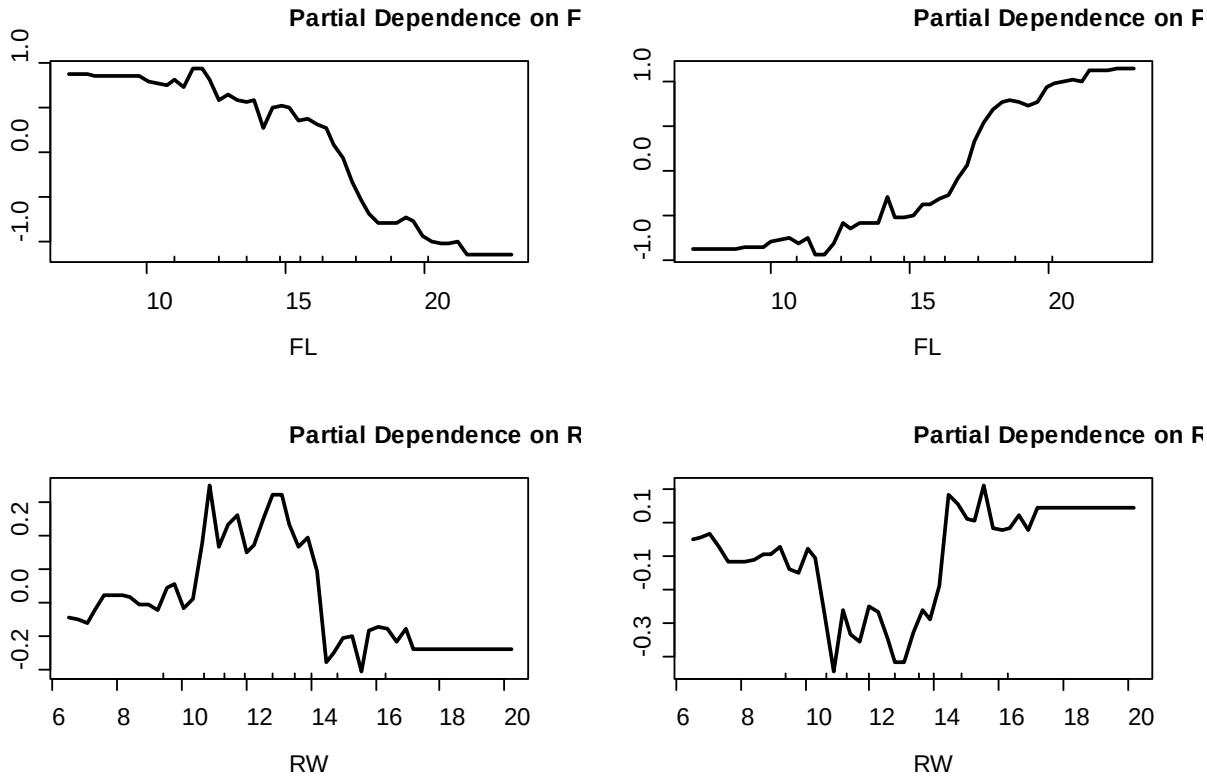
```
> set.seed(2)
> varUsed(randomForest(sp ~ FL + RW + CL + CW + BD, data=crabs, ntree=100,
mtry =3, by.tree=TRUE, count=TRUE))

[1] 455 214 311 543 432
```

Nejvyšší frekvenci výběru mají tři nejvýznamnější proměnné.

Z předešlých výsledků jsme získali představu o důležitosti proměnných, teď zbývá zjistit jejich chování v kategoriích. K tomu poslouží funkce *partialPlot*, která zobrazí graf efektu proměnné na pravděpodobnost kategorie. Do funkce zadáme výsledný les, proměnnou a kategorii (pomocí parametru *which.class*), pro kterou chceme znázornit průběh proměnné.

```
> set.seed(123)
> par(mfrow=c(3,2))
> partialPlot(les_krabi, crabs, FL, which.class= "B", cex.axis=1.2,
cex.lab=1.2, lwd=2)
> partialPlot(les_krabi, crabs, FL, which.class= "O", cex.axis=1.2,
cex.lab=1.2, lwd=2)
```

…atd.



Výsledné hodnoty jsou zobrazeny pro modrou variantu kraba (B) vlevo, pro oranžovou variantu (O) vpravo. Na ose *y* jsou znázorněny změny pravděpodobnosti, na ose *x* hodnoty proměnné. Znázorněné proměnné FL (velikost čelního laloku) a RW (šířka zadečku) mají vyšší hodnoty pro oranžovou variantu. Pokud bychom vykreslili grafy i ostatních proměnných, zjistíme, že všechny mají podobný průběh, budou tedy korelované a navzájem zástupné. Pro klasifikaci by stačilo použití proměnné, která má nejvyšší významnost. Dvě varianty kraba bychom tedy měli být schopni dobře odlišit i na základě pouze jedné proměnné a to buď velikosti čelního laloku, výšce trupu nebo šířce krunýře (v sestupném pořadí podle jejich významnosti). Pokud je velikost čelního laloku menší než 15 mm, jedná se o modrou variantu, hodnoty přibližně od 17 mm výše o variantu oranžovou. Nižší hodnoty ve změně pravděpodobnosti nastávají u proměnné RW (šířka zadečku) a také CL (délka krunýře), což vysvětluje jejich malou významnost.

Podobnou informaci můžeme získat z prototypů kategorií založených na matici těsnosti pomocí funkce *classCenter*. Získáme tak odhad hodnot proměnných, které jsou charakteristické pro danou barevnou variantu. Prototyp je medoid[11] z jeho nejbližších sousedů (z „reprezentativních“ pozorování) z příslušné kategorie. Můžeme porovnat hodnoty prototypů pro obě varianty.

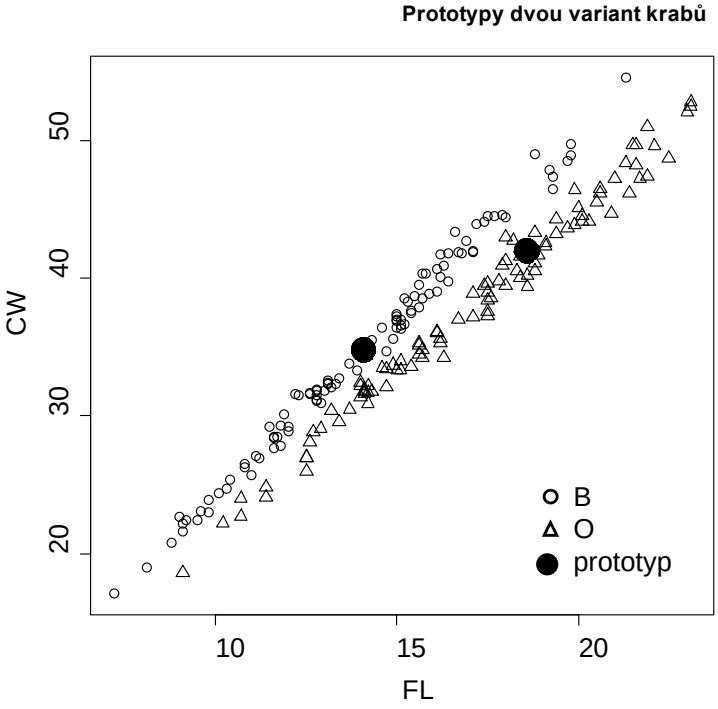
```
> set.seed(22)
> les_krabi <- randomForest(sp ~ FL+RW+CL+CW+BD, data=crabs, mtry=3,
ntree=150, importance=TRUE, prox=TRUE)
> krabi_prototyp <- classCenter(crabs[,4:8], crabs[,1], les_krabi$prox)
> krabi_prototyp

    FL    RW    CL   CW   BD
B 14.1 11.85 30.05 34.8 12.3
O 18.6 14.60 37.80 42.0 17.2
```

[11] Obecněji medoid charakterizuje střed shluku objektů, přičemž jeho průměrná vzdálenost k ostatním objektům v tomto shluku je minimální.

Zde se nám potvrdily naše závěry z předešlých grafů efektu proměnných na predikci. Prototypy jsou u všech proměnných vyšší pro oranžovou variantu kraba. Samozřejmě jednoduchým výpočtem mediánů nebo průměrů bychom došli ke stejnému závěru. Zde jsou však tyto hodnoty spočítány z nejpodobnějších vzorků a informace o charakteristické hodnotě proměnné je více jednoznačná. Jinými slovy, krab s hodnotami prototypů pro modrou variantu bude opravdu modrý.

Zobrazíme ještě vztah dvou nejvýznamnějších proměnných a jejich prototypů:

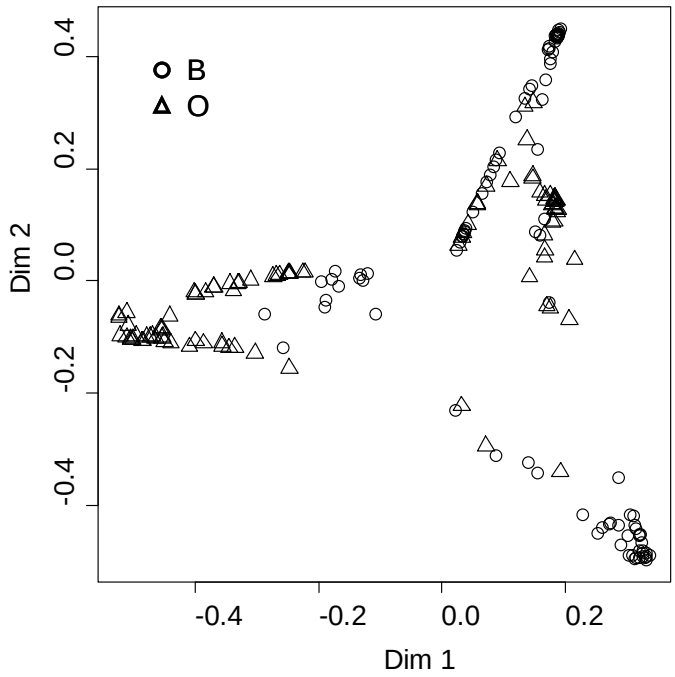
```
> plot(crabs[,4], crabs[,7], pch=21, xlab=names(crabs)[4],
ylab=names(crabs)[7],cex.axis=1.5, cex=1.5, cex.lab=1.5, bg=c("red",
"blue")[as.numeric(factor(crabs$sp))], main="Prototypy dvou variant krabů")
> points(krabi_prototyp[,1], krabi_prototyp[,4], pch=21, cex=3, bg=c("red",
"blue"))
```



Proměnné velikost čelního laloku a šířka krunýře jsou vysoce korelované. Rozdělení na dvě varianty je však patrné.

Kromě výpočtu prototypů použijeme matici těsnosti ve spojení s vícerozměrnými metodami, konkrétně ve Vícerozměrném škálování (MDS). Nastavíme funkci pro grafické znázornění první a druhé faktorové osy *MDSplot* s argumenty našeho optimálního lesa a závisle proměnné pro definici kategorií.

```
> MDSplot(les_krabi, crabs$sp, palette=rep(1,2), pch=as.numeric(crabs$sp),
cex.axis=1.5, cex=1.5, cex.lab=1.5)
```



Vzorky z obou kategorií se částečně překrývají (zejména v pravém horním rohu). Bylo by zajímavé zjistit, zda by rozdělení barevné varianty podle pohlaví mohlo odlišit tyto vzorky.

Podíváme se, jak se liší samičky a samečci nezávisle na barevné variantě.

```
> set.seed(51)
> les_krabi4 <- randomForest(sex ~ FL+RW+CL+CW+BD, data=crabs, mtry=3,
ntree=100, importance=TRUE, prox=TRUE)
> les_krabi4

Call:
 randomForest(formula = sex ~ FL + RW + CL + CW + BD, data = crabs,
 mtry = 3, ntree = 100, importance = TRUE, prox = TRUE)
               Type of random forest: classification
                     Number of trees: 100
No. of variables tried at each split: 3

        OOB estimate of  error rate: 10%
Confusion matrix:
   F  M class.error
F 89 11        0.11
M  9 91        0.09
```

Vidíme, že rozlišení krabů podle pohlaví je stejně úspěšné jako podle barevné varianty.

```
> importance(les_krabi4, type=2)
```
/typ 1 a 2

```
   MeanDecreaseGini       MeanDecreaseAccuracy
FL         7.958217 FL            0.7952684
RW        47.836443 RW            1.3953118
CL        20.190412 CL            1.2057500
CW        11.942759 CW            0.9773177
BD        11.470370 BD            1.0503910
```

Nejvýznamnější proměnnou pro odlišení pohlaví krabů je šířka zadečku (RW) (tato proměnná byla pro klasifikaci do dvou barevných variant bez odlišení pohlaví nevýznamná),

vzrostl také vliv proměnné délky krunýře (CL). Parametry se tak informačně doplňují, mají vliv buď na určení barevné varianty, nebo na pohlaví.

```
> krabi_prototyp1 <- classCenter(crabs[,4:8], crabs[,2], les_krabi4$prox)
> krabi_prototyp1

    FL    RW    CL    CW   BD
M 15.00 11.50 31.40 34.20 13.6
F 16.55 14.35 33.95 38.85 14.7
```

Analýzou prototypů jednoduše zjistíme, že samičky jsou ve všech ohledech větší než samečci. Rozdíly však vzhledem k rozsahu proměnných nejsou nijak velké.

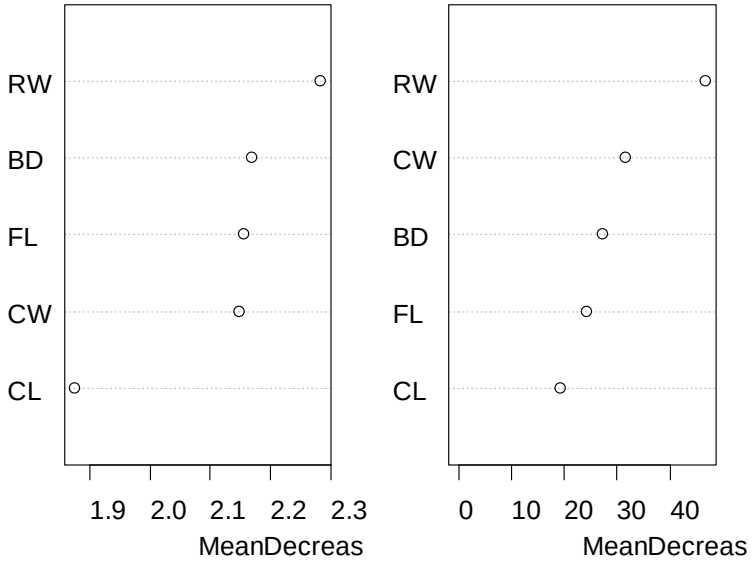
Logickým pokračováním je nyní rozdělení souboru na čtyři kategorie: kombinace pohlaví a barevné varianty.

```
> set.seed(321)
> les_krabi1 <- randomForest(TX$spsex ~ FL+RW+CL+CW+BD, data=crabs,
mtry=3, ntree=150, importance=TRUE, prox=TRUE)
> les_krabi1

Call:
 randomForest(formula = TX$spsex ~ FL + RW + CL + CW + BD, data = crabs,
mtry = 3, ntree = 150, importance = TRUE, prox = TRUE)
               Type of random forest: classification
                     Number of trees: 150
No. of variables tried at each split: 3

        OOB estimate of  error rate: 18.5%
Confusion matrix:
   BF BM OF OM class.error
BF 43  2  4  1        0.14
BM  2 42  0  6        0.16
OF  4  0 41  5        0.18
OM  2  7  4 37        0.26
```

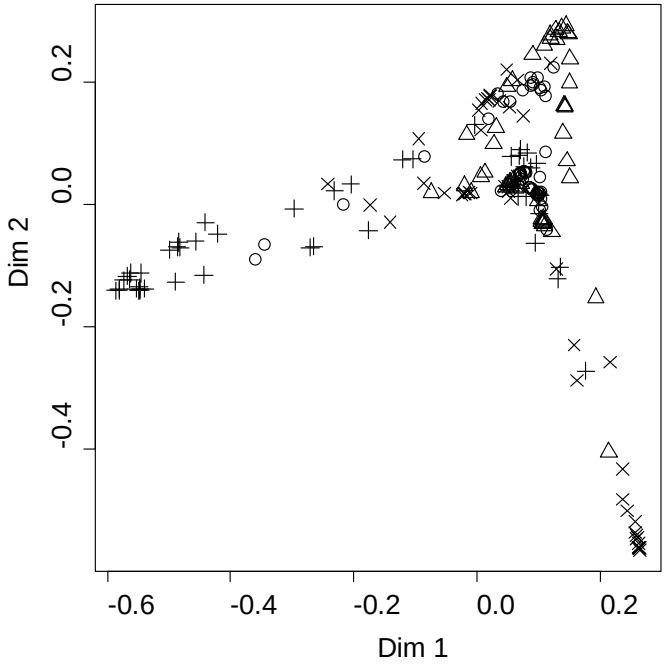
Klasifikace do čtyř skupin podle varianty a pohlaví dopadla hůře, než klasifikace pouze do dvou kategorií. O něco větší překryv nastává mezi samičkami z obou barevných variant. Hodnoty prototypů proměnných rovněž potvrzují hypotézu podobnosti samiček obou variant, mezi samečky je větší rozdíl zejména v šířce zadečku (RW) a délce krunýře (CL).

```
> krabi_prototyp1

     FL   RW   CL   CW   BD
BM 14.6 11.3 31.4 36.4 13.2
BF 14.9 13.2 30.1 35.6 13.0
0M 18.8 13.7 39.1 43.2 17.8
OF 19.1 16.3 37.9 42.8 17.2
```

```
> varImpPlot(les_krabi1, cex =1.4)
```



Pořadí významnosti proměnných se změnilo, mezi nejvýznamnější proměnné patří šířka zadečku a šířka krunýře. Opět bychom mohli zkoumat efekt proměnných na klasifikaci a prototypy jednotlivých kategorií. Pro ostatní klasifikace je rovněž nutné znovu otestovat parametry *ntree* a *mtry*.



Vraťme se zpátky k prvním dvěma osám z vícerozměrného škálování. Z grafu jsou patrné dvě odlišné skupiny, jsou to samičky modré (+) a samečci oranžové (x) varianty. Tento výsledek není překvapivý, protože hodnoty prototypů ukázaly největší rozdíl právě mezi těmito skupinami.

Nyní se podíváme, jak lze doplnit chybějící hodnoty pomocí náhodného lesa. K tomu je určena funkce *rfImpute*. Protože náš datový soubor žádné prázdné hodnoty neobsahuje, musíme

nejdříve nějaké vytvořit. Dvacet vzorků z celkového souboru obsahujícího 200 pozorování nahradíme hodnotou *NA*.

```
> krabi_chybi <- crabs
> set.seed(111)
> for (i in 4:8) krabi_chybi[sample(200, sample(20)), i] <- NA
> krabi_chybi

    sp sex index   FL   RW   CL   CW   BD
1    B   M     1  8.1   NA 16.1 19.0  7.0
2    B   M     2  8.8  7.7 18.1 20.8  7.4
3    B   M     3  9.2  7.8 19.0 22.4  7.7
4    B   M     4  9.6  7.9 20.1 23.1  8.2
5    B   M     5  9.8  8.0 20.3 23.0  8.2
6    B   M     6 10.8  9.0 23.0 26.5  9.8
7    B   M     7 11.1   NA 23.8 27.1  9.8
8    B   M     8 11.6  9.1 24.5 28.4 10.4
9    B   M     9 11.8  9.6 24.2 27.8  9.7
10   B   M    10 11.8 10.5 25.2 29.3   NA
11   B   M    11 12.2 10.8 27.3 31.6 10.9
12   B   M    12   NA 11.0 26.8 31.5 11.4
…….
```

Defaultní nastavení funkce *rfImpute* je pro pět iterací a 300 stromů. Nyní doplníme chybějící hodnoty na základě měření proximity:

```
> krabi_dopln <- rfImpute(sp ~ FL+RW+CL+CW+BD, data=krabi_chybi, iter=5,
ntree=300)

ntree      OOB      1      2
  300:  15.00% 13.00% 17.00%
ntree      OOB      1      2
  300:  12.50% 13.00% 12.00%
ntree      OOB      1      2
  300:  11.50% 13.00% 10.00%
ntree      OOB      1      2
  300:  12.00% 12.00% 12.00%
ntree      OOB      1      2
  300:  12.50% 12.00% 13.00%
```

Výsledkem jsou hodnoty klasifikační chyby *oob* vzorků pro pět iterací.

Podíváme se, jak se změní procento chyby klasifikace, po doplnění 10% chybějících hodnot, od klasifikace bez chybějících hodnot.

```
> set.seed(15)
> randomForest(sp ~ FL + RW + CL + CW + BD, data = krabi_dopln,
importance = TRUE, proximity = TRUE, ntree = 150, mtry =3)

Call:
 randomForest(formula = sp ~ FL + RW + CL + CW + BD, data = krabi_dopln,
 importance = TRUE, proximity = TRUE, ntree = 150, mtry = 3)
               Type of random forest: classification
                     Number of trees: 150
No. of variables tried at each split: 3

        OOB estimate of  error rate: 13.5%
Confusion matrix:
   B  O class.error
B 88 12        0.12
O 15 85        0.15
```

Celková chyba vzrostla pouze o 3%. O něco hůře dopadla kategorie *O*, jejíž chybovost vzrostla o 5%. V praxi však toto srovnání nemáme, a proto je dobré se vždy rozhodnout, zda je lépe chybějící vzorky (a tím i celé řádky!) oželet nebo se pokusit o doplnění. Pokud je doplníme, měli bychom mít na paměti, že celková chybovost může být ovlivněna právě doplněnými hodnotami. Nejlépe je provést klasifikaci na menším souboru bez chybějících hodnot a následně s doplněním na celém souboru a porovnat výsledky obou klasifikací.

Poslední velmi užitečná funkce, kterou si vyzkoušíme, je pro predikci nových vzorků a jmenuje se příhodně *predict*. Opět můžeme nastavit parametry pro zobrazení výsledku. Parametr *type* nabízí tři možnosti: *response* – na výstupu budou uloženy predikované hodnoty, *prob* – výstupem bude matice pravděpodobností jednotlivých kategorií a *vote* – matici hlasování jednotlivých stromů. Dalším parametrem je *norm.votes*, pokud je nastaven na TRUE, počty hlasů budou vyjádřeny jako podíl kategorií (normované), jinak bude výstupem přímo počet hlasů. Nastavením parametru *predict.all* na *TRUE* bude uložena predikce všemi stromy, jinak pouze výsledné hlasování nebo průměrování.

Nejdříve si soubor rozdělíme na dva podsoubory, jeden použijeme pro tvorbu lesa a druhý soubor (s odstraněním závisle proměnné) použijeme pro predikci. Vytvoříme proměnnou s hodnotami 1 a 2 pro rozdělení souboru, poměr rozdělení udává parametr *prob*. V tomto případě rozdělíme soubor na 80% a 20% původního souboru.

```
> set.seed(112)
> index <- sample(2, nrow(crabs), replace = TRUE, prob=c(0.8, 0.2))
> krabi_lesP<- randomForest(sp ~ FL + RW + CL + CW + BD, data=crabs[index
== 1,], mtry=3, ntree = 150)
> krabi_pred <- predict(krabi_lesP, crabs[index == 2,], type="response",
norm.votes=TRUE, predict.all=FALSE, proximity=FALSE)
> krabi_pred

18  25  40  44  48  55  60  65  71  78  82  92  97  99 100 103 105 110 113 114
  B   B   B   B   B   B   B   B   O   B   B   B   B   O   O   B   O   O   O   O
115 118 123 127 128 129 134 139 141 142 151 157 168 171 181 190 193 198 200
  O   O   O   O   O   O   O   O   B   O   B   O   O   O   O   O   O   O   O
Levels: B O
```

Výsledek predikce můžeme zobrazit v tabulce:

```
> table(observed = crabs[index==2, "sp"], predicted = krabi_pred)

        predicted
observed  B  O
       B 12  3
       O  3 21
```

Dále si můžeme zobrazit další varianty výsledků, predikce jednotlivými stromy a matici těsnosti (*proximity*) nebo pravděpodobnost zařazení všech pozorování do kategorie B a O:

```
> predict(krabi_lesP, crabs[index == 2,], predict.all=TRUE,
proximity=TRUE)
> krabi_pred <- predict(krabi_lesP, crabs[ind == 2,], norm.votes = FALSE)
> krabi_pred <- predict(krabi_lesP, crabs[ind == 2,], type="prob")

          B           O
18  0.993333333 0.006666667
25  0.980000000 0.020000000
40  0.813333333 0.186666667
44  0.573333333 0.426666667
```

```
48  0.780000000 0.220000000
55  0.973333333 0.026666667
60  0.880000000 0.120000000
65  0.766666667 0.233333333
71  0.306666667 0.693333333
78  0.700000000 0.300000000
82  0.820000000 0.180000000
....
```

Z pravděpodobností jednotlivých vzorků můžeme vybrat vzorky jednoznačně zařazené lesem (např. 18, 25, 55) a vzorky, u kterých zařazení do dané kategorie není tak snadné (44).

Tento příklad měl sloužit k demonstraci možností náhodného lesa. K podobným závěrům bychom došli i za použití jiných metod, jako například ANOVA, diskriminační analýza či logistická regrese. Představíme-li si však soubor se stovkami proměnných, různých rozložení a typů, jsou výhody této metody zřejmé.

5.2 Další typy lesů

Lesů je opět celá řada, na závěr zmiňme ještě dva typy: Bagging a Boosting.

Bagging (*bootsrap aggregating*) [18] používá k vytvoření lesa, stejně jako Random Forests, boostrapové výběry [27]. Soubor se rozdělí na trénovací a testovací. Trénovací se použije k tvorbě stromu a testovací ke zjištění predikční síly. Počet náhodných výběrů udává počet stromů v lese. Výsledek je opět dán buď většinovým hlasováním, nebo průměrnou hodnotou ze všech stromů. Rozdíl oproti náhodným lesům v tvorbě jednotlivých stromů je ten, že nejsou náhodně vybírány prediktory, ale pouze pozorování. Bagging se tak dá považovat za zjednodušenou verzi Random Forests, u které není ošetřena korelace mezi stromy.

Boosting (zesilování) pracuje podobně jako bagging, les je tvořen větším množstvím stromů a výsledek, zařazení pozorování do kategorie, je dán většinovým hlasováním rozhodovacích stromů. Každému pozorování je však přiřazena určitá váha, podle toho jak dobře jej lze klasifikovat. V prvním kroku je všem pozorováním přiřazena stejná váha, následně se datový soubor upraví podle výsledku klasifikace a větší váha se přiřadí pozorováním, které dopadly hůře. Tímto způsobem se dá výrazně zvýšit přesnost a predikční síla modelu. Nicméně je třeba říci, že lze takto „natrénovat“ lesy i na pozorování, která mohou být odlehlá a stromy se pak snaží vysvětlit i šum v datech. Toto nebezpečí nastává zejména u souborů s velmi velkou variabilitou. Boosting byl vytvořen pro klasifikační problém, ale lze jej rozšířit i na regresi. Populárním algoritmem pro boosting je AdaBoost.M1 [28], znám také jako Discrete AdaBooost [29].

Příloha: Úvod do programu R

(převzato a upraveno z výukových materiálů E. Gelnarové - Analýza dat na PC III: Pokročilý SW pro analýzu dat)

Charakteristika:

- nekomerční, volně šiřitelný statistický software
- objektově orientovaný jazyk
- ovládání z příkazové řádky, otevřený zdrojový kód
- možnost tvorby vlastních procedur
- pod Unix i Windows
- ke stažení na adrese: **http://cran.at.r-project.org/** (zde zdrojový i zkompilovaný kód)
- dodatečné knihovny také na adrese: **http://www.bioconductor.org/** [cit. 2012-1-31]

Instalace:

- klasicky (setup.exe)
- doinstalování dalších knihoven: zkompilovanou knihovnu nakopírovat do adresáře …/*library*, nebo přidat přes *Packages -> Instal package(s) from CRAN* (kurzívou je označen výběr z menu na horní liště v prostředí Windows)

Vybrané knihovny:

- ***lattice*** - nástroje pro vizualizaci vícerozměrných dat; obsahuje soubor *environmental* (příklad III a VI)
- ***rpart*** - knihovna pro klasifikační a regresní stromy
- ***MASS*** - knihovna obsahující velké množství souborů včetně *Titanic* (příklad V) a *crabs* (příklad XI) a funkcí jako cvičení ke knize: W. N. Venables, B. D. Ripley: Modern Applied Statistics with S (4th edition, 2002)
- ***CHAID*** - knihovna pro výpočet stromu CHAID - není ve standardní nabídce instalace R, je však možné ji nainstalovat pomocí příkazu:
  `>install.packages("CHAID", repos="http://R-Forge.R-project.org")`
- ***datasets*** – knihovna obsahuje velké množství souborů pro testování různých metod včetně souboru *trees* (příklad VIII)
- ***prim*** – knihovna pro výpočet metod PRIM
- ***earth*** – knihovna pro výpočet metod MARS
- ***randomForest*** – knihovna pro výpočet metody Random Forest

Přivolání knihovny:

- seznam knihoven: *help -> Packages* /knihovnu už musíme mít nainstalovanou
- `> library (nazevknihovny)` - např. `> library(MASS)`
  (Zobáček `> je na` úvodu příkazové řádky. Příkaz na příkazovou řádku je uvozen tímto znakem. Tento znak není součástí příkazu.)
- nebo volíme: *packages -> Load Package*

Nápověda:

- *Help -> Html Help -> Packages + Search Engine & Keywords, Search Engine* vyžaduje Javu!
- Spuštění hypertextového helpu z příkazové řádky: `> help.start()` informace o konkrétní proceduře - příkazový řádek: např: `> help(median)`
- vypsání zdrojového kódu procedury: pouze název procedury např. `> median`
- kompletní manuály ke stažení na http://www.r-project.org/

Začínáme s prací:

- Nastavení pracovního adresáře (zde se ukládají a vyhledávají soubory dat a výsledků): *File -> Change directory* nebo z příkazového řádku: `> setwd(“.....”)`
- Zjištění aktuálního nastavení pracovního adresáře: `> getwd(“.....”)`
- Načtení vlastní procedury ve formátu procedura.r : *File -> Source R code* nebo z příkazového řádku: `> source(file)`

Načtení dat:

- **datový formát soubor.Rdata:**
  File -> Load WorkSpace
  nebo z příkazového řádku:
  `> load(file, envir = parent.frame()),`
  `> load("C:/Rko/glass.Rdata")`(příklad IX)
- **soubor z načtené knihovny:**
  `> data(environmental)`
- **textový soubor soubor.txt**
  data ve sloupcích, sloupce oddělené mezerou, sloupce nemají jména:
  `> read.table(file, header = FALSE, sep = "", quote = "\"'", dec = “.“,…)`
  data ve sloupcích, sloupce oddělené mezerou, sloupce mají jména:
  `> read.table("mammals1.txt",header=TRUE,sep=" ",dec=".")`
- **textový soubor, další předdefinované formáty:**
  ```
  > read.csv(file, header = TRUE, sep =",", quote="\"", dec=".", ...)
  > read.csv2(file, header = TRUE, sep =";", quote="\"", dec=",", ...)
  > read.delim(file, header = TRUE, sep ="\t", quote="\"", dec=".",...)
  > read.delim2(file, header = TRUE, sep ="\t", quote="\"", dec=",",...)
  ```
- **data uložena v Clipboardu:**
  `> read.table("clipboard",header=TRUE)` – načte tabulku včetně hlaviček, kterou jsme uložili do clipboardu (CTRL-C)

Načtení skriptů:

- tvorba v externím textovém editoru (*Notepad, WinEdit, Tinn-R,....)*, přípona *.r*, nebo také v rámci R.
- tvorba v rámci R: *File -> New script (open script)*
- samostatné procedury
- spouštění částí kódu: pravé tlačítko myši + *Run line or selection* (nebo Ctrl-R)

Tvorba objektů + práce s objekty:

- názvy proměnných - rozlišují se velká a malá písmena (*X*, *x* - dva různé objekty)
- přiřadíme názvu proměnné hodnotu: `> x<-4`
- přiřadíme názvu proměnné hodnotu: `>z<-x` nebo také
  `> z<-x^2+abs(z)/cos(2)+pi-exp(1)*log(8)+log10(8)/sqrt(10)`
- vytvoření vektoru: `> v<-c(1,2,3,5,6,9)`
- *k*-tý prvek vektoru: `> v[k]`
- vytvoření matice 3x4, která má všechny prvky nulové: `> matice<-matrix(0,3,4)`
- prvek v prvním řádku a druhém sloupci je překvapivě nula: `> m[1,2]`
- posloupnost od jedné do deseti, skok = 0,5: `> posloupnost<-seq(1,10,by=0.5)`
- uchop objekt: `> attach(objekt)`
- vymaž objekt: `> rm(objekt)`
- vypiš seznam existujících objektů : `> ls()`
- vypiš seznam existujících objektů + některé další informace : `> ls.str()`

Datové typy:

- typ numerický (speciální numerické hodnoty*: Inf, -Inf, NaN = Not A Number*)
- typ znakový ("Toto je řetězec")
- typ komplexní
- typ logický (TRUE, FALSE)
- typ libovolného objektu (např. Oo) zjistíme pomocí funkce: `>mode(Oo)`
- chybějící hodnoty jsou reprezentovány kódem: ***NA = Not Available***

Datové struktury:

- vektor (*vector*), matice (*matrix*), pole (*array*), datová tabulka (*data frame*), seznam (*list*), faktor (*factor*)
- typ datové struktury libovolného objektu (např. Oo) zjistíme pomocí funkce: `>is(Oo)`
- Konkrétní dotaz: `>is.numeric(Oo)`, podobně pro ostatní datové struktury

Počítání s vektory:

- s vektory lze použít stejné operace jako s čísly (mají-li vektory stejnou délku!)
- příklad BMI:
```
> weight<-c(60,72,57,90,95,72)
> height<-c(1.75,1.8,1.65,1.9,1.74,1.91)
> bmi<-weight/height^2
> bmi
```
- zjištění délky vektoru: `>length(weight)`

Elementární vektorové funkce:

- součet: `> sum(weight)`
- minimum a maximum: `> min(weight), max(weight)`
- průměr a medián: `> mean(weight), median(weight)`
- rozptyl a kvantily: `> var(weight), quantile(weight,c(0.3,0.5))`
- kumulativní součet: `> cumsum(weight)`

Práce s maticemi:

- tvorba matice z vektoru: např. `> matice<-matrix(c(1,2,3,4,5,6),2,3)`
- matice lze transponovat: `> maticetr<-t(matice)`
- matice lze násobit (pozor na rozměry!): `> matice%*%maticetr,` sčítat a násobit skalárem...atd.
- submatice 2x2: `> sub<-matice[1:2,1:2]`
- spojování matic - "slepujeme sloupce": `> cbind(maticeA,maticeB)`
- spojování matic - "slepujeme řádky": `> rbind(maticeA,maticeB)`

Pole:

- třírozměrné pole: `> pole<-array(1:24,c(3,2,4))`
- zjištění dimenze pole: `> dim(pole)`
- prvek pole (analogicky jako v případě vektoru a matice): `> pole[1,2,3]`
- vrstva pole (analogicky jako v případě matice): `> pole[1,,]`

Obrázky a grafy:

- kreslíme graf: příkaz *plot* má velké množství parametrů pro nastavení: barvy, bodů, popisu os... etc.
- příklad:
```
> s<-seq(-3,3,by=0.1)
> d<-dnorm(s)
> plot(s,d, main="Hustota N(0,1) a T(5)",col="red", type="p",
xlab="kvantily")
```
- do stávajícího grafu dokreslujeme další funkce:
```
> t<-dt(s,5)
> lines(s,t,type="l",col="blue")
```
- uložení obrázku: *file -> save as…*
- pro listování mezi obrázky je nutné průběžné ukládání:
```
> history -> recording
```
- histogram
```
> x<-rnorm(1000,0,1)
> hist(x)
```
- box-plot
```
> boxplot(x, notch=TRUE)
> y<-rnorm(1000,1,1)
> r<-c(x,y)
> nula<- matrix(0,1,1000)
> jedna<-matrix(1,1,1000)
> group<-c(nula,jedna)
> group<-c(rep(0,1000),rep(1,1000))
> boxplot(r~group, col="yellow")
```

Chybějící pozorování:

- Chybějící pozorování jsou reprezentována kódem *NA*.
```
> tNA<- krabi_chybi (příklad XI)
> attach(tNA)
> is.na(RW)
> sum(RW)
> sum(RW,na.rm=TRUE)
```

Konec práce a uložení obrázků:

- výpis objektů, které máme momentálně v pracovním prostoru: `> ls()`
- ukončení práce s programem: `> q()`

Uložení obrázků:

```
> savePlot(filename = "Rplot", type = c("wmf", "emf", "png", "jpeg",
"jpg", "bmp", "ps", "eps", "pdf"), device = dev.cur(), restoreConsole =
TRUE)
```

Uložení dat:

- ve formátu .Rdata:
  `> save(data, file="data.Rdata")`
- jako textový soubor (množství parametrů stejné jako při načítaní dat):
  ```
  > write.table(data, file="data.txt",append=FALSE)
  > write.table(data, file="data.txt",append=TRUE)
  > write.csv2(data, file= ="data.csv”)
  ```

Další literatura a učební texty:

- Michal Kulich: R pro samouky (česky)
  http://www.karlin.mff.cuni.cz/~kulich/vyuka/Rdoc/uvodrfpm.pdf [cit. 2012-1-31]
- + odkazy na další anglicky psané texty
  http://www.karlin.mff.cuni.cz/~kulich/vyuka/Rdoc/index.html [cit. 2012-1-31]
- R publikace
  http://www.r-project.org/doc/bib/R-publications.html [cit. 2012-1-31]
  A mnoho dalších… (Springer: edice use R!)

Seznam použité literatury

[1] Schnoor, J.L., Environmental Modeling: Fate of Chemicals in Water, Air and Soil, John Wiley & Sons, New York (1996)

[2] Guisan, A., Zimmermann, N.E.: Predictive habitat distribution models in ecology. Ecological Modelling135, 147--186 (2000)

[3] Legendre, P., Legendre, L.: Numerical Ecology. 2nd Engl. ed., Elsevier, Amsterdam (1998)

[4] Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Chapman and Hall, New York (1984)

[5] Elsner, J.B., Lehmiller, G.S., Kimberlain, T.B.: Objective Classification of Atlantic Hurricanes. J. Climate 9, 2880--2888 (1996)

[6] Hastie, T., Tibshirani, R., Friedman, J.H.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd ed. Springer, New York (2009)

[7] Bruntz, S.M., Cleveland, W.S., Kleiner, B., Warner, J.L.: The Dependence of Ambient Ozone on Solar Radiation, Wind, Temperature, and Mixing Height. In: Symposium on Atmospheric Diffusion and Air Pollution, pp. 125--128. American Meterological Society, Boston (1974)

[8] Cleveland, W.S.: Visualizing Data. Hobart Press, Summit, New Jersey (1993)

[9] Quinlan, J.R.: Induction of Decision Trees. Mach. Learn. 1, 1, 81--106 (1986)

[10] Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann (1993)

[11] Loh, W.Y., Shih, Z.S.: Split Selection Methods for Classification Trees. Statistica Sinica 7, 815--840 (1997)

[12] Kass, G.V.: An Exploratory Technique for Investigating Large Quantities of Categorical Data. Applied Statistics 29, 119--127 (1980)

[13] Dawson, R.J., Mac, G.: The 'Unusual Episode' Data Revisited. Journal of Statistics Education, 3. http://www.amstat.org/publications/jse/v3n3/datasets.dawson.html [cit. 2011-06-22]

[14] Friedman, J.H., Fisher, N.I.: Bump-hunting in High-dimensional Data. Statistics and Computing 9, 123--143 (1999)

[15] Friedman, J.H.: Multivariate Adaptive Regression Splines. Annals of Statistics 19, 1--141 (1991)

[16] Atkinson, A.C.: Plots, Transformations and Regression. Oxford University Press (1985)

[17] Breiman, L.: Machine Learning: Wald I. (2002) http://www.stat.berkeley.edu/users/Breiman/RandomForests/cc papers.html [cit. 2011-10-18]

[18] Breiman, L.: Bagging Predictors. Machine Learning 24, 123--140 (1996)

[19] Breiman, L.: Random Forests. Machine Learning 45, 5--32 (2001)

[20] Venables, W.N., Ripley, B.D.: Modern Applied Statistics in S. Springer, 4th edition (2002)

[21] Looking Inside the Black Box, Wald II. (2002) http://www.stat.berkeley.edu/users/Breiman/RandomForests/cc papers.html [cit. 2011-10-18]

[22] Breiman, L.: Interface '04 Short Course. (2004) http://www.stat.berkeley.edu/users/Breiman/RandomForests/cc papers.html [cit. 2011-10-18]

[23] Culter, A.: Interface '04 Short Course. (2004) http://www.stat.berkeley.edu/users/Breiman/RandomForests/cc papers.html [cit. 2011-10-18]

[24] Kubošová, K., Brabec, K., Jarkovský, J., Syrovátka, V.: Selection of Indicative Taxa for River Habitats: a Case Study on Benthic Macroinvertebrates using Indicator Species Analysis and the Random Forest Methods. Hydrobiologia 651, 101--114 (2010)
[25] Campbell, N.A., Mahon, R.J.: A Multivariate Study of Variation in Two Species of Rock Crab of Genus Leptograpsus. Australian Journal of Zoology 22, 417--425 (1974)
[26] Venables, W.N., Ripley, B.D.: Modern Applied Statistics with S. 4th edition. Springer (2002)
[27] Efron, B., Tibshirani, R.J.: An Introduction to the Bootstrap. Chapman and Hall, London (1993)
[28] Freund, Y., Schapire, R.: A Decision-theoretic Generalization of Online Learning and an Application to Boosting. Journal of Computer and System Sciences 55, 119--139 (1997)
[29] Friedman, J., Hastie T., Tibshirani, R.: Additive Logistic Regression: a Statistical View of Boosting (with Discussion). Annals of Statistics 28, 337--407 (2000)

Summary

The text was created as a study material for the Advanced Non-parametric Methods lectures under the EFS project No. CZ.1.07/2.2.00/07.0318 Multidisciplinary Innovation Of Study In Mathematical Biology. The publication is intended primarily for students of Mathematical Biology and further to other interested persons from various natural science fields.

The focus of the study material is on tree based methods (TBM). TBM are multi-dimensional techniques used for regression analysis and classification. These techniques are very useful in various disciplines for solving real problems.

The first chapter covers basic concepts such as individual steps of the modeling process, types of variables and the difference between classification and regression analyses. Subsequent chapters explain TBM algorithms such as the Classification and Regression Trees (CART), Chi-squared Automatic Interaction Detector (CHAID), Patient Rule Induction Method (PRIM) and Multivariate Adaptive Regression Splines (MARS). Tree topology, criterial statistics, stability of the trees and validation mechanisms are described, too. Chapter ends are dedicated to the advantages and disadvantages of individual tree types. The last chapter is focused on the Random Forest method which is an extension of the CART decision trees.

The theory is extended by practical examples in the R program. By practicing these examples, students gain experience with learning and validating the individual methods.

The emphasis is put primarily on understanding the methods principles, their interpretation and correct application.

Obsah

Rozhodovací stromy a lesy
Mgr. Klára Komprdová, Ph.D.

Recenzenti: Doc. RNDr. Lubomír Popelínský, PhD.; Mgr. David Zelený, Ph.D.
Obálka: Radim Šustr, DiS
Jazyková korekce: Ing. Marie Juranová
Ilustrace: Lucie Paceltová
Vydalo: AKADEMICKÉ NAKLADATELSTVÍ CERM, s.r.o. Brno,
Purkyňova 95a, 612 00 Brno
www.cerm.cz
Tisk: FINAL TISK s.r.o. Olomučany
Náklad: 200 ks
Vydání: první
Vyšlo v roce 2012

ISBN 978-80-7204-785-7