13 Moderní metody výběru příznaků ve statistickém rozpoznávání

13.1 Úvod

Obor *rozpoznávání obrazů* či prostě *rozpoznávání* může být s určitým zjednodušením charakterizován jako řešení problému klasifikace či reprezentace dat popisujících zkoumané objekty reálného světa pomocí vektorů příznaků. Redukce dimenzionality (dimensionality reduction DR) je prováděna buď v podobě extrakce, nebo selekce (výběru) příznaků, která optimalizuje nějaké vhodné kritérium. Výběr příznaků je jedním z klíčových postupů předzpracování dat používaný při řešení nejrůznějších úloh ve statistickém rozpoznávání, strojovém učení, zpracování obrazové informace, klasifikaci dokumentů dobývání znalostí z rozsáhlých databází atd. Smyslem redukce dimenzionality je nejen ušetřit čas a prostor vyřazením nepodstatných částí dat, ale i zlepšit úspěšnost klasifikace či přesnosti reprezentace dat v budovaném systému potlačením vlivu šumových či jinak neinformativních příznaků.

13.2 Redukce dimenzionality

Termínem „obraz“ budeme označovat D-rozměrný reálný vektor $\boldsymbol{x} = (x_1,..., x_D) \in \mathsf{X} \subseteq \mathsf{R}^D$, jehož prvky jsou měření odpovídající vlastnostem reprezentovaného objektu. Vektor $\boldsymbol{x}$ budeme rovněž nazývat vektorem příznaků. Příznaky jsou veličiny specifikované pro daný problém odborníkem. Na počátku specifikované příznaky by měly pokrýt maximum zjistitelné informace o uvažovaných objektech – možnou nadbytečnou informaci lze dodatečně redukovat pomocí metod popisovaných v této kapitole. Informaci od počátku chybějící však nahradit nelze. Ve většině praktických úloh proto očekáváme vysokou vstupní dimenzionalitu.

V kontextu statistického rozpoznávání nejčastěji předpokládáme, že objekt reprezentovaný obrazem $\boldsymbol{x}$ má být klasifikován do jedné z konečného počtu C různých tříd $\Omega = \{\omega_1,\ldots, \omega_C\}$. Obraz $\boldsymbol{x} \in \mathsf{X}$ patřící do třídy ω_i považujeme za realizaci náhodného vektoru vybranou náhodně v souladu s apriorními pravděpodobnostmi $P(\omega_i)$ a podmíněnými hustotami pravděpodobnosti $p(\boldsymbol{x}|\omega_i)$ $\omega_i \in \Omega$.

Smyslem redukce dimenzionality je tedy nalézat automaticky d nových příznaků na základě vstupních D měření (pokud možno $d \ll D$) tak, aby bylo maximalizováno (či minimalizováno) vhodné kritérium informativnosti příznaků.

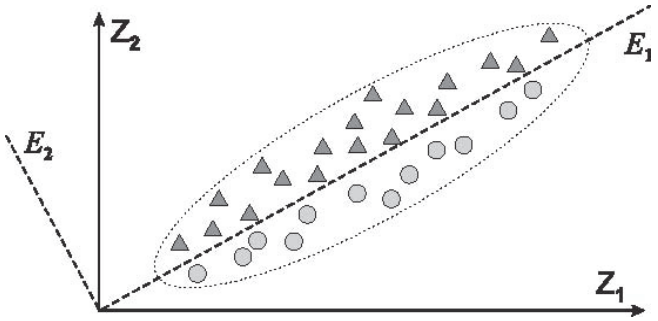
13.2.1 Redukce dimenzionality podle charakteru výsledných příznaků

Z praktických důvodů rozlišujeme dva základní přístupy k DR:
– redukci dimenzionality pomocí extrakce příznaků (feature extraction, FE),
– redukci dimenzionality pomocí výběru příznaků (feature selection, FS).
Extrakcí příznaků rozumíme proces, při němž je výsledný vektor získán jakoukoliv transformací z vektoru původního. Výsledné příznaky proto mohou reprezentovat kombinaci informace všech původních příznaků a mohou být zcela odlišně interpretovatelné. *Výběrem příznaků* rozumíme speciální případ extrakce, kdy je jednoduše vybrána podmnožina původních příznaků. Ačkoliv extrakce v obecnosti umožňuje přesnější reprezentaci dat v podprostoru, výběr bývá často upřednostněn z praktických důvodů. Je jednodušší, zachovává původní význam a interpretovatelnost měření a umožňuje úspory nákladů při akvizici dat (např. v medicíně je vhodné rozeznat, která měření netřeba u pacientů nadále provádět, aniž by se zhoršila úspěšnost automatického diagnostického systému). V dalším výkladu se zaměříme výhradně na problém *výběru příznaků*.

13.2.2 Redukce dimenzionality podle cíle

Alternativně lze rozlišit dva přístupy k redukci dimenzionality podle cíle:
– redukci dimenzionality pro optimální reprezentaci dat,
– redukci dimenzionality pro klasifikaci.

V prvním případě je cílem co nejlépe zachovat v méně rozměrném prostoru informaci obsaženou v původních datech. V druhém případě je cílem maximalizovat rozlišitelnost různých tříd.



Obr. 13.1 Redukce dimenzionality pro optimální reprezentaci dat nemusí být optimální pro rozlišení tříd.

Na obrázku 13.1 ilustrujeme, že výsledek DR pro optimální reprezentaci dat nemusí být vhodný pro rozlišení tříd. Kolečka a trojúhelníky reprezentují vzorky dvou různých tříd v dvojrozměrném prostoru. Aplikujeme-li na tento příklad metodu hlavních komponent – principal component analysis, PCA, viz např. (Duda a kol., 2000), budou data v jednorozměrném prostoru optimálně reprezentována osou E_1. Tato metoda DR vhodná pro reprezentaci však není vhodná pro klasifikaci, neboť pro odlišení tříd je v tomto případě optimální osa E_2.

V dalším textu se soustředíme především na problém klasifikace. Pro tento účel lze DR provádět *lokálně*, tj. pro každou jednotlivou třídu zvlášť, nebo *globálně*, tj. společně

pro všechny třídy. Pro obsáhlejší přehled problematiky odkazujeme na práce (Duda a kol., 2000), (Webb, 2002), (McLachlan, 2004), (Ripley, 2005) a (Theodoridis a kol., 2006).

13.3 Výběr podmnožiny příznaků

Je dána množina X_D o D příznacích. Označme X_d množinu všech možných podmnožin množiny X_D kardinality d, kde d reprezentuje požadovaný počet příznaků. Nechť $J(X)$ je kriteriální funkce, která vyhodnocuje podmnožinu příznaků $X \in \mathsf{X}_d$. Bez ztráty obecnosti předpokládejme, že větší hodnota kritéria J indikuje lepší podmnožinu příznaků. Potom lze FS problém zformulovat tak že najdeme podmnožinu $\tilde{X}_d$, pro kterou platí

$$J(\tilde{X}_d) = \max_{X \in \mathsf{X}_d} J(X) \,. \tag{13.1}$$

Za předpokladu, že je k dispozici vhodná funkce J k ohodnocování efektivity podmnožin příznaků, je problém výběru příznaků redukován na vyhledávací problém podmnožiny maximalizující dané kritérium. Poznamenejme, že volba počtu příznaků d může být složitým problémem závisejícím na charakteru úlohy, pokud není optimalizace hodnoty d součástí vyhledávacího procesu.

Proces výběru příznaků probíhá ve čtyřech základních krocích: *vytváření podmnožin*, *ohodnocování podmnožin*, *ukončení procesu na základě zvoleného stop-kritéria* a *ověřování výsledků*. Volba vhodné strategie a kritéria pro ohodnocení kvality podmnožin příznaků jsou základními faktory při návrhu algoritmů pro výběr příznaků. Podrobný přehled různých aspektů procesu výběru příznaků je uveden v publikaci (Liu Yu, 2005).

Dosavadní výzkum a naše vlastní zkušenosti nás dovedly k závěru, že neexistuje jediný vhodný obecně aplikovatelný přístup k problému výběru příznaků. Různé existující metody jsou vhodné v různých specifických situacích a selhávají v situacích jiných. Volba konkrétní metody (a kritéria) závisí vždy na znalosti konkrétního problému. Z tohoto důvodu je vývoj nových metod a příslušných paradigmat předmětem neustálého zájmu.

13.3.1 Volba metod výběru příznaků podle optimality

Metody výběru příznaků lze rozdělit na dvě skupiny podle záruky optimality výsledku:

Optimální metody. Tyto metody zahrnují například *úplné prohledání*, které je vhodné pouze pro úlohy o nízké dimenzionalitě, a *akcelerační metody* často založené na principu větví a mezí (Branch & Bound), jež jsou schopny účinně zredukovat počet testovaných kombinací d příznaků, kladou však omezující podmínky na použité kritérium. Všechny optimální metody lze považovat za značně pomalé pro řešení problémů o vysoké dimenzionalitě, viz oddíl 13.4.

Suboptimální metody. Tyto metody jsou kompromisem mezi rychlostí vyhledávání a optimalitou řešení. Zahrnují mimo jiné individuální vyhodnocování příznaků, nejrůznější gradientní metody, randomizované i deterministické, popř. kombinované, metody, viz např. (Devijver a kol., 1982), (Pudil a kol., 1994a), (Hussein a kol., 2001)

a (Somol a kol., 2008b). Ačkoliv u suboptimálních metod není zaručena optimalita výsledku, jejich výhody obvykle nad tímto nedostatkem značně převažují. Optimalita výsledku vzhledem ke zvolenému kritériu navíc nemusí mít přímý vztah k úspěšnosti klasifikace finálního klasifikátoru (Raudys, 2006). Pro vysokodimenzionální problémy jsou suboptimální metody s ohledem na výpočetní složitost jedinou volbou, viz oddíl 13.5.

13.3.2 Volba metod výběru příznaků podle způsobu vyhodnocování kritéria

S ohledem na způsob ohodnocování testovaných podmnožin příznaků můžeme metody výběru příznaků rozdělit do dalších skupin:

Filtry (*Filters*), viz (Devijver kol., 1982), (Kohavi a kol., 1997), (Dash a kol., 2002) a (Yu a kol., 2003), používají k ohodnocení podmnožiny příznaků jako kritéria míry vzdálenosti informační míry, míry závislosti a míry konzistence, jejichž hodnoty jsou vypočteny přímo použitím trénovacích dat, viz oddíl 13.5.

Pouzdra (*Wrappers*), viz (Kohavi a kol., 1997), vyžadují předem učící algoritmus – jeho účinnost je přímo kritériem pro ohodnocení kvality vybrané podmnožiny. Obecně mají tyto metody oproti filtrům větší účinnost pro daný učící algoritmus, ale jsou výpočetně složitější a jimi vybrané příznaky mohou být špatné v obecnějším kontextu, viz oddíl 13.5.

Integrované metody (*Embedded methods*), viz (Guyon a kol., 2003), ale také (Kononenko, 1994) nebo (Pudil a kol., 1995) a (Novovičová a kol. 1996), integrují proces výběru příznaků do procesu odhadu modelu. Návrh modelu a výběr příznaků jsou tudíž neoddělitelné učící procesy, na které je možné pohlížet jako na speciální tvar metody pouzder. Integrované metody tedy nabízejí účinnost, jež konkuruje metodě pouzder, umožňují rychlejší proces učení, ale dávají výsledky úzce svázané s konkrétním modelem, viz také oddíl 13.8.

Hybridní metody (*Hybrid methods*), viz (Das, 2001), (Sebban a kol., 2002) (Liu a kol., 2005) a (Somol a kol., 2006), kombinují uvedené metody za účelem dosažení co nejlepší účinnosti určitého učícího algoritmu (např. klasifikátoru) při nízké časové složitosti srovnatelné s filtry, viz oddíl 13.7.

13.3.3 Volba metod výběru příznaků podle znalosti problému

Alternativně lze volit metody výběru příznaků podle *apriorní* znalosti pravděpodobnostních charakteristik problému:

Základní informace je známa – neboli je možné alespoň předpokládat, že podmíněné hustoty pravděpodobnosti jsou unimodální. V takovém případě můžeme postupovat odhadem parametrů vhodného modelu a následnou maximalizací některé pravděpodobnostní míry vzdálenosti (Mahalanobisovy, Bhattacharyyovy atd.) za pomoci některé optimální či suboptimální metody, obvykle typu filtr (Devijver a kol., 1982).

Žádná informace není známa – neboli jediným zdrojem informace jsou trénovací data samotná. Nemůžeme předpokládat unimodalitu podmíněných hustot pravděpodobnosti, nebo naopak předpokládáme jejich komplikovanou strukturu. V takovém případě může být výhodné zvolit vhodnou metodu typu pouzdro, ale v některých případech (zejména při dostatečné velikosti dat) je možné využít alternativní metody výběru příznaků založené na modelování struktury dat za použití směsi hustot pravděpodobnosti speciálního typu. Směsový model umožňuje zachytit velmi přesně vlastnosti dat a impli-

kuje též formu klasifikačního pravidla. O směsových metodách diskutujeme v oddílu 13.8.

13.4 Optimální vyhledávací metody

Problém *optimálního* výběru příznaků (jako i obecnosti hledání optimální podmnožiny) je velmi obtížný, především vzhledem k výpočetní náročnosti. Vyhledání optimální podmnožiny příznaků mezi všemi podmnožinami dané kardinality je kombinatorický problém nezvládnutelný úplným prohledáním všech konfigurací již při relativně malém počtu uvažovaných příznaků. Již delší dobu je proto tomuto problému věnována značná pozornost ve snaze vyhledávací proces urychlit. Urychlení bylo doposud dosaženo různými způsoby, a to buď za cenu uvolnění podmínky optimality řešení (Devijver a kol., 1982), (Kirkpatrick a kol., 1983), (Caruana a kol., 1994), (Jain a kol., 1997), (Chaikla a kol., 1999) a (Kudo a kol., 2000), nebo definicí různých pomocných heuristik umožňujících identifikovat takové části vyhledávacího prostoru, které mohou být z vyhledávání vyloučeny bez ohrožení optimality výsledku (Devijver a kol., 1982) a (Fukunaga, 1990). Mezi metodami schopnými redukovat prostor hledání bez ztráty optimality lze za nejdůležitější označit algoritmus Branch & Bound (algoritmus větví a mezí) a algoritmy od něj odvozené. Idea algoritmu větví a mezí je dobře známa i mimo kontext výběru příznaků a je považována za jeden ze základních nástrojů v oblasti umělé inteligence (Lawler a kol., 1966), (Nilsson, 1971) (Kumar a kol., 1983), (Mitschele-Thiel, 1994), (Webb, 1995), (Nilsson, 1998) a (Korf, 1999). Poprvé byl algoritmus použit pro výběr příznaků v práci (Narendra a kol., 1977). V dalších letech byl v tomto kontextu detailněji zkoumán a doplňován (Fukunaga, 1990), (Hamamoto a kol., 1990), (Yu a kol., 1993), (Kudo a kol., 2000), (Chen, 2003) (Somol a kol., 2004) a (Nakariyakul a kol., 2007).

Rodina algoritmů větví a mezí řeší problém optimálního výběru d příznaků z původní množiny D příznaků za předpokladu, že kriteriální funkce J splňuje podmínku monotónnosti. Buď $\overline{\chi}_j$ množina příznaků vzniklá odebráním j příznaků $y_1, y_2, \ldots, y_j$ z množiny X_D všech D příznaků neboli

$$\overline{\chi}_j = X_D \setminus \{y_1, y_2, ..., y_j\}, \quad j = 1, 2, ..., D-1 .$$

Podmínka monotónnosti vyžaduje, aby pro podmnožiny příznaků $\overline{\chi}_1, \overline{\chi}_2, \ldots \overline{\chi}_j$, kde

$$\overline{\chi}_1 \supset \overline{\chi}_2 \supset ... \supset \overline{\chi}_j ,$$

kriteriální funkce J splňovala

$$J(\overline{\chi}_1) \geq J(\overline{\chi}_2) \geq ... \geq J(\overline{\chi}_j) . \tag{13.2}$$

Využitím podmínky monotónnosti je možné zabránit průchodu některými částmi vyhledávacího prostoru, a tím zredukovat počet vyhodnocování kriteriální funkce, a tedy zkrátit potřebný výpočetní čas. Ačkoliv má i v nejhorším případě algoritmus větví a mezí exponenciální časovou náročnost, v praxi rychlostí zásadně překonává úplné vyhledávání. Značné úsilí bylo v posledních letech věnováno pokusům o dodatečná

zrychlení tohoto hledacího schématu. Byla navržena řada modifikovaných verzí algoritmu přinášejících zrychlení nejčastěji pro různé specifické problémy (Foroutan a kol., 1987), (Mitschele-Thiel, 1994), (Koller a kol., 1996) a (Liu a kol., 1998). Algoritmus byl paralelizován (Gengler a kol., 1994), (Xu a kol., 1995), (Yang a kol., 1999) a (Iamnitchi a kol., 2000). Zároveň byl zkoumán sám pojem optimality výběru příznaků, což vedlo k rozlišení hledacích algoritmů na *filtry* a *pouzdra* (Kohavi a kol., 1997), viz též odst. 13.3.2. Nejvýraznějšího urychlení algoritmu větví a mezí bylo v poslední době dosaženo zavedením pomocného predikčního mechanismu, který dovoluje nahradit řadu pomocných výpočtů hodnot kritéria jednoduchým odhadováním hodnot přibližných, aniž by ovšem byla ohrožena optimalita celkového výsledku (Somol a kol., 2004). Dvěma moderním algoritmům z této skupiny se budeme podrobněji věnovat v odstavcích 13.4.3 a 13.4.4.

Tabulka 13.1 Hlavní charakteristiky diskutovaných algoritmů optimálního výběru příznaků

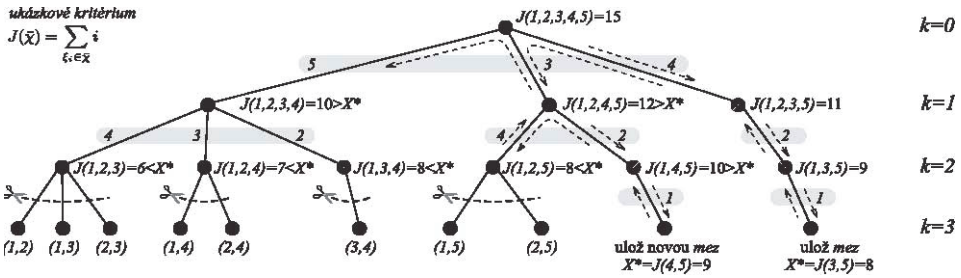
Algoritmus větví a mezí (Branch & Bound)		*Horizontální řazení uzlů stromu*	*Ušetření výpočtů při průchodu větví*
základní	(BBB, odst. 13.4.1)	NE	NE
vylepšený	(IBB, odst. 13.4.1)	výpočtem	NE
s částečnou predikcí	(BBPP, odst. 13.4.3)	predikcí	NE
rychlý	(FBB, odst. 13.4.4)	predikcí	predikcí

13.4.1 Základní pojmy a varianty algoritmu větví a mezí

Přibližme nejprve základní princip všech vyhledávacích algoritmů větví a mezí. Algoritmus konstruuje vyhledávací strom, v němž kořen reprezentuje množinu všech příznaků D a listy vyjadřují podmnožiny požadované cílové kardinality d. Procházeje stromem od kořene k listu, algoritmus postupně odebírá po jednom příznaku z momentální množiny „kandidátů“ ($\overline{\chi}_k$ v k-té úrovni stromu). Algoritmus udržuje informaci o doposud nejlepší (v listu) nalezené podmnožině $\mathcal{X}^*$ a odpovídající hodnotě kritéria X^* (tato hodnota je onou *mezí* [bound] z názvu algoritmu). Kdykoliv v některém vnitřním uzlu stromu klesne aktuální hodnota kritéria pod tuto mez, podmínka monotónnosti (13.2) dovolí odříznout celý podstrom daného uzlu, v němž již nemůže dojít ke zvýšení meze neboli k nalezení lepšího řešení. Tímto způsobem je ušetřena značná část výpočtů spojených s vyhodnocováním kriteriální funkce. Běh algoritmu větví a mezí je demonstrován na obrázku 13.2 pro ukázkový problém výběru $d = 2$ příznaky z celkového počtu $D = 5$. Čárkované šipky naznačují postup průchodu stromem. Pro podrobnější výklad odkazujeme na práce (Devijver a kol., 1982), (Fukunaga, 1990) a (Narendra a kol., 1977).

Je známa řada vylepšení tohoto základního schématu. Každý vyhledávací strom může být zredukován na „minimální vyhledávací strom“ (Yu a kol., 1993) vynecháním nevětvících se uzlů neboli zkrácením cest. Pro jednoduchost tento mechanismus nezahrnujeme do detailních popisů algoritmů, byl však vzat v úvahu při experimentech. Za obecně nejpoužívanější variantu algoritmu lze doposud považovat tzv. „vylepšený“ algoritmus větví a mezí, na který se budeme odkazovat podle původního názvu IBB (Improved Branch & Bound) (Fukunaga 1990). Bez nároku na přesnost nazvěme *poklesem hodnoty kritéria* rozdíl mezi momentální hodnotou kritéria a hodnotou po odebrání jednoho příznaku. Nazvěme *špatným příznakem* takový příznak, jehož odebrání z pracovní množiny kandidátů způsobí pouze *mírný pokles* hodnoty kritéria. Obdobně

označme za *dobrý příznak* takový, jehož odebrání způsobí *značný pokles* hodnoty kritéria. (V tuto chvíli není třeba vymezovat přesně pojmy mírný a značný.) Je zřejmé, že i při neměnné topologii výpočetního stromu lze definovat různá konkrétní přiřazení příznaků hranám stromu. Algoritmus IBB se snaží přiřadit *špatné* příznaky do pravé, tj. řidší části stromu, a *dobré* příznaky do levé, tedy hustší části stromu. Smysl tohoto řazení je v urychlení *růstu meze* v průběhu výpočtu. Nejprve jsou totiž v pravé části stromu zkoumány cílové podmnožiny získané odebíráním těch *špatných* příznaků z X_D, které způsobí co nejmenší pokles hodnoty kritéria. Rychlejší růst meze v počátečním stadiu výpočtu v kombinaci s umístěním *dobrých* příznaků v levé části stromu vede v pozdějších fázích výpočtu k častějšímu poklesu testovaných hodnot kritéria pod mez a tím k efektivnějšímu ořezání hustých větví.



Obr. 13.2 Příklad problému řešeného algoritmem větví a mezí, kde $d = 2$ optimální dvojice příznaků je vybírána z celkového počtu $D = 5$ příznaků tak, aby maximalizovala ilustrativní uměle definovanou kriteriální funkci.

Efekt tohoto heuristického *horizontálního řazení uzlů* je ilustrován na obr. 13.2. Ukázaný strom neodpovídá stromu „vylepšeného" algoritmu IBB protože první úroveň stromu není vhodně seřazena (sekvence příznaků 5, 3, 4 zvýrazněná šedivým pozadím). Změníme-li pořadí tak, aby odpovídalo IBB pořadí (5, 4, 3), algoritmus položí mez rovnu skutečnému optimu hned při prvním vyhodnocení listu a tím ušetří o jedno vyčíslení kritéria více, $J(4, 5)$. Všimněte si, že v zakřížkovaných vrcholech je vyhodnocování kritéria nadbytečné, poněvadž tyto vrcholy leží na cestě. Přeskočením těchto vrcholů zredukujeme výpočetní strom na již zmíněný „minimální vyhledávací strom". Algoritmus IBB v kombinaci s „minimálním vyhledávacím stromem" budeme v dalším výkladu považovat za referenční podobu algoritmu větví a mezí.

13.4.2 Nevýhody tradičních algoritmů větví a mezí

Každý algoritmus větví a mezí provádí řadu výpočtů, které mohou, ale také nemusí, vyústit v celkovou časovou úsporu a které by nebyly potřeba při využití algoritmu úplného prohledávání. Hodnota kritéria není počítána pouze pro cílové podmnožiny d příznaků $\bar{\chi}_{D-d}$, ale navíc také pro jejich nadmnožiny $\bar{\chi}_{D-d-j}$ $j = 1, 2,..., D - d$.

Princip algoritmu větví a mezí nezaručuje, že bude možné odříznout dostatek větví výpočetního stromu a že tedy celková doba výpočtu bude kratší než v případě algoritmu úplného prohledání. Skutečná rychlost výpočtu závisí vždy na vlastnostech použité kriteriální funkce i na konkrétních zpracovávaných datech. Teoreticky nejhorší situaci mů-

žeme ilustrovat nadefinováním umělé kriteriální funkce $J(\overline{\chi}_k) = |\overline{\chi}_k| \equiv D - k$ (viz obr. 13.6a). Tato funkce způsobí vyhodnocování v každém uzlu výpočetního stromu a vede tedy celkově ke zhruba dvojnásobnému počtu výpočtů, než by vyžadoval algoritmus úplného prohledávání. V případě úplného prohledávání je předem známo, kolik výpočtů bude provedeno, v případě algoritmu větví a mezí celkový počet výpočtů přesně predikovat nelze.

Nebezpečí příliš neefektivního běhu algoritmu větví a mezí vyplývá mimo jiné z těchto pozorování:

a) výpočty jsou tím pomalejší, čím blíže kořeni stromu jsou prováděny (zpracovávají se větší podmnožiny příznaků),

b) pravděpodobnost odříznutí větve je blíže kořeni naopak menší (hodnoty kritéria vnitřních uzlů jsou blíže kořeni větší, a tím spíše neklesnou pod dosavadní mez počítanou v listech).

Klasické algoritmy větví a mezí takto spotřebují většinu výpočetního času na pomalé vyhodnocování málo perspektivních vnitřních uzlů výpočetního stromu. Tento efekt je znatelný zejména pro $d \ll D$. Ve „vylepšeném“ algoritmu IBB je navíc potřeba významné množství dodatečných výpočtů k heuristickému řazení uzlů stromu, jak uvidíme v následujícím textu.

13.4.3 Vylepšení „vylepšeného“ algoritmu

Ukažme si nejprve zásadní nevýhodu „vylepšeného“ algoritmu IBB z obrázku 13.2. Když algoritmus konstruuje první úroveň stromu, neboli když určuje pořadí následníků kořene, tak vyhodnocuje *pokles hodnoty kritéria* pro všech 5 momentálně použitelných příznaků z množiny doposud nepřiřazených příznaků Ψ. Pouze 3 příznaky jsou však poté použity, výsledky zbylých dvou výpočtů se dále nepoužijí.

Naším cílem je nalézt stejné (nebo nepříliš odlišné) řazení uzlů stromu jako v případě „vylepšeného“ algoritmu IBB avšak s menším počtem vyhodnocování kritéria. K tomuto účelu použijeme jednoduchý predikční mechanismus. Další úrovně stromu budeme konstruovat ve dvou fázích. V první fázi pouze rychle odhadneme *poklesy hodnoty kritéria* pro všechny příznaky z Ψ. Příznaky seřadíme podle odhadnutých hodnot a pro konstrukci stromu použijeme pouze nutný počet příznaků q_k v pořadí od *největší hodnoty* poklesu. Pouze pro tyto vybrané příznaky poté dopočteme skutečné hodnoty kritéria a na jejich základě popřípadě poopravíme pořadí. Tímto způsobem lze snížit celkový počet vyhodnocování kritéria při konstrukci jednotlivých úrovní stromu na úroveň základního algoritmu větví a mezí, výsledný strom se však nebude příliš lišit od efektivně seřazeného stromu IBB.

K predikci používáme statistiku *poklesů hodnot kritéria* získanou v průběhu dosavadního výpočtu zvlášť pro každý jednotlivý příznak. Nazvěme $\boldsymbol{A} = [A_1, A_2,\ldots, A_D]^T$ *vektorem příspěvků* příznaků k hodnotě kritéria. Tento vektor bude pro každý příznak zaznamenávat průměrný pokles hodnoty kritéria způsobený odebíráním tohoto příznaku v různých okamžicích výpočtu. Počet vyhodnocování poklesu hodnoty kritéria pro jednotlivé příznaky budeme udržovat ve *vektoru čítačů* $\boldsymbol{B} = [B_1, B_2,\ldots, B_D]^T$. Pro formální popis tohoto algoritmu větví a mezí s částečnou predikcí (Branch & Bound with Partial Prediction BBPP) respektujeme konvence značení z knihy (Devijver a kol., 1982):

k – úroveň stromu ($k = 0$ značí kořen),

$\overline{\chi}_k = \{\xi_j \mid j = 1, 2, ..., D-k\}$ – momentální pracovní množina „kandidátských“ příznaků v k-té úrovni stromu,

q_k – počet následníků zpracovávaného uzlu (v další úrovni stromu),

$Q_k = \{Q_{k\,1}\ Q_{k\,2}\ \dots\ Q_{k,q_k}\}$ – seřazená množina příznaků přiřazených hranám vedoucím k následníkům právě zpracovávaného uzlu (všimněte si, že „kandidátské“ podmnožiny $\overline{\chi}_{k+1}$ jsou jednoznačně určeny pomocí příznaků $Q_{k\,i}$, kde $i = 1,\dots, q_k$),

$\boldsymbol{J}_k = [J_{k,1}, J_{k,2},\dots, J_{k,q_k}]^{\mathrm{T}}$ – vektor hodnot kritéria odpovídajících následníkům právě zpracovávaného uzlu $J_{k,i} = J(\overline{\chi}_j \setminus \{Q_{k,i}\}$ pro $i = 1, ..., q_k)$,

$\Psi = \{\psi_j \mid j = 1, 2,\dots, r\}$ – řídicí množina udržující r příznaků momentálně dostupných pro konstrukci další úrovně stromu, neboli pro sestavení množiny Q_k, množina Ψ tak zaručuje optimální topologii stromu,

$\mathscr{J} = \{x_j \mid j = 1, 2,\dots, d\}$ – nejlepší doposud nalezená podmnožina d příznaků,

X^* – momentální mez (hodnota kritéria odpovídající množině $\mathscr{J}$).

Poznámka: Hodnoty q_j množiny Q_j a vektory $\boldsymbol{J}_j$ musí být vždy uloženy pro všechna $j = 0,\dots, k$, aby byl umožněn zpětný průchod stromem (backtracking).

Kdykoli v průběhu výpočtu dojde v souvislosti s odebráním nějakého příznaku y_i z nějaké aktuální „kandidátské“ podmnožiny $\overline{\chi}_k$ k výpočtu hodnoty kritéria $J(\overline{\chi}_k \setminus \{y_i\})$, bude následovat aktualizace *predikční informace*

$$A_{y_i} = \frac{A_{y_i} B_{y_i} + (J(\overline{\chi}_k) - J(\overline{\chi}_k \setminus \{y_i\})).1}{B_{y_i} + 1} \tag{13.3}$$

a

$$B_{y_i} = B_{y_i} + 1, \tag{13.4}$$

kde A_{y_i} i B_{y_i} jsou na začátku inicializovány hodnotou 0 pro všechna $i = 1,\dots, D$.

Algoritmus větví a mezí s částečnou predikcí (BBPP)

Inicializace: $k = 0$, $\overline{\chi}_0 = X_D$, $\Psi = X_D$, $r = D$, X^* udává nejnižší strojově možnou hodnotu.

KROK 1: *Zvolme následníky právě zpracovávaného uzlu, a tím vytvořme další úroveň stromu*. Nejprve určeme jejich počet $q_k = r - (D - d - k - 1)$. Nyní sestavme uspořádanou množinu Q_k a vektor $\boldsymbol{J}_k$ takto: seřaďme všechny příznaky $\psi_j \in \Psi$, $j = 1,\dots, r$, sestupně podle jim odpovídajících hodnot A_{ψ_j}, $j = 1, ..., r$, neboli

$$A_{\psi_1} \geq A_{\psi_2} \geq ... \geq A_{\psi_{j_r}},$$

a vyberme postupně prvních q_k z nich. Položme

$$Q_{k,i} = \text{ pro } i = 1,..., q_k \quad \text{a} \quad J_{k,i} = J(\overline{\chi}_k \setminus \{\psi_{j_i}\}) \text{ pro } i = 1, ..., q_k.$$

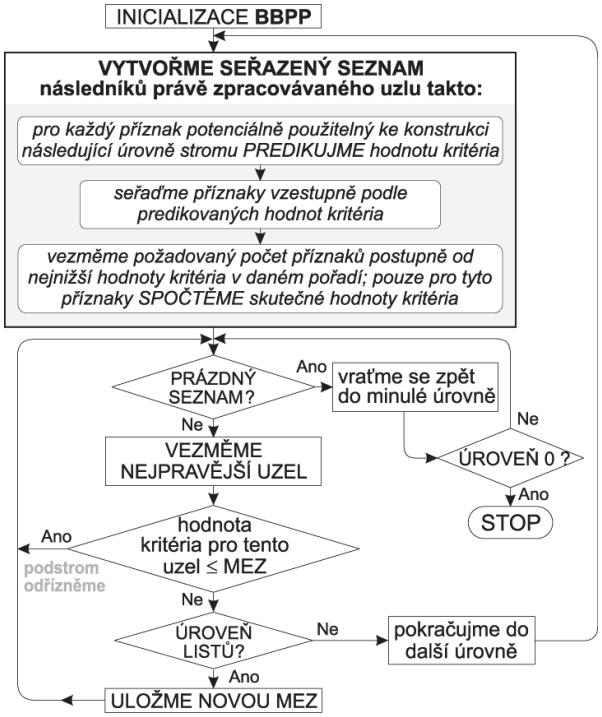
Abychom zabránili budoucím redundantním vyhodnocováním kritéria, vylučme příznaky ψ_{j_i} z další konstrukce stromu, neboli položme $\Psi = \Psi \setminus Q_k$ a $r = r - q_k$.

KROK 2: *Otestujme nejpravějšího následníka (připojeného hranou Q_{k,q_k})*. Pokud $q_k = 0$, potom všichni následníci již byli otestováni a proto pokračujme Krokem 4 (backtracking). Pokud $J_{k,q_k} < X^*$ pokračujme Krokem 3. Jinak položme $\overline{\chi}_{k+1} = \overline{\chi}_k \setminus \{Q_{k.q_k}\}$. Pokud $k + 1 = D - d$, pak byl dosažen list, a proto pokračujme Krokem 5. Jinak pokračujme průchodem do další úrovně stromu: položme $k = k + 1$ a jděme na Krok 1.

KROK 3: *Následný uzel připojený hranou Q_{k,q_k} (včetně celého jeho podstromu) může být odříznut*. Vraťme příznak Q_{k,q_k} do množiny příznaků určených k další konstrukci stromu, neboli položme $\Psi = \Psi \cup \{Q_{k,q_k}\}$ a $r = r + 1$, $Q_k = Q_k \setminus \{Q_{k,q_k}\}$ a $q_k = q_k - 1$ a pokračujme jeho levým sousedem, jděme na Krok 2.

KROK 4: *Návrat do předchozí úrovně*. Položme $k = k - 1$. Pokud $k = -1$, pak celý strom byl již zkonstruován a algoritmus může být ukončen. Jinak vraťme příznak Q_{k,q_k} do množiny „kandidátů“. Položme $\overline{\chi}_k = \overline{\chi}_{k+1} \cup \{Q_{k,q_k}\}$ a jděme na Krok 3.

KROK 5: *Aktualizujme hodnotu meze.* Položme $X^* = J_{k,q_k}$. Uložme doposud nejlepší nalezenou cílovou podmnožinu $\mathscr{X} = \overline{\chi}_{k+1}$ a jděme na Krok 2.



Obr. 13.3 Zjednodušený diagram algoritmu větví a mezí s částečnou predikcí (BBPP).

O vlastnostech algoritmu BBPP diskutujeme v odstavcích 13.4.5 a 13.4.6. Zjednodušený diagram algoritmu naleznete na obr. 13.3. Poznamenejme ještě, že formální popis BBPP se od IBB liší pouze v Kroku 1, kde by v případě IBB bylo místo prostého seřazení predikovaných hodnot $A_{\psi j}$ nutné před řazením nejprve spočítat všechny skutečné hodnoty $J(\bar{\chi}_k \setminus \{\psi_j\})$.

13.4.4 Rychlý algoritmus větví a mezí

Tak zvaný rychlý algoritmus větví a mezí (Fast Branch & Bound FBB) (Somol a kol., 2004) byl navržen s cílem ještě podstatněji zredukovat počet vyhodnocování kritéria ve vnitřních uzlech výpočetního stromu. K tomu účelu bylo dalekosáhleji využito predikčního mechanismu. Algoritmus se snaží využít znalost dosavadních poklesů hodnoty kritéria nejen k řazení uzlů, ale i při průchodu stromem do hloubky. Skutečné i predikované hodnoty jsou při konstrukci stromu využívány ekvivalentním způsobem s výjimkou dvou specifických situací:

1. predikce nemá smysl v úrovni listů a
2. testování predikované hodnoty oproti mezi nikdy nemůže vést samo o sobě k odříznutí podstromu.

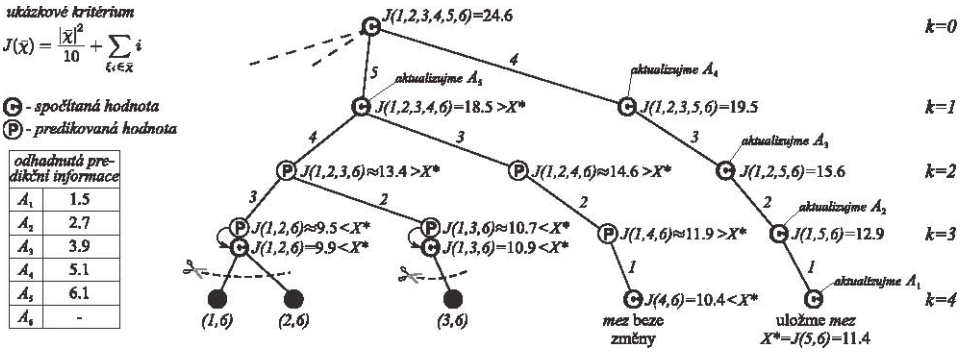
Zůstává-li v průběhu výpočtu predikovaná hodnota kritéria znatelně větší než *momentální mez*, je možné předpokládat, že ani skutečná hodnota nejspíše nepadá pod mez a odpovídající podstrom nelze odříznout. V takové situaci algoritmus jednoduše pokračuje konstrukcí další úrovně stromu. Přiblíží-li se ovšem v průběhu výpočtu predikovaná hodnota momentální mezi, popř. pod tuto mez klesne, možnost odříznutí podstromu je třeba ověřit dopočtením skutečné hodnoty kritéria. Pouze když se i skutečná hodnota ukáže nižší než mez, je podstrom algoritmem odříznut. Všimněme si, že tímto způsobem zůstává zaručena optimalita výsledného řešení, ačkoliv v řadě vnitřních uzlů stromu vůbec nedošlo k výpočtu skutečné hodnoty kritéria. Všimněme si dále, že ani případné nepřesnosti či chyby predikce optimalitu výsledku neovlivní, mohou pouze vést k průchodu částí stromu, která by za jiných okolností mohla být odříznuta. V sekci experimentů ukážeme, že ani v případě velmi výrazných chyb predikčního mechanismu neklesá efektivita algoritmu na úroveň algoritmů nevyužívajících predikce.

V rámci algoritmu FBB je nutné uchovávat informaci o tom, která hodnota kritéria v různých částech stromu byla získána tím kterým způsobem. K tomu využijeme *typového vektoru*, v němž pro každý uzel právě zpracovávané úrovně stromu bude zaznamenán symbol „P“ nebo „C“ podle toho, byla-li hodnota kritéria v uzlu predikována či skutečně vyčíslena. Informace o typu uzlu především rozlišuje, je-li případnou možnost odříznutí podstromu nutné nejdříve ověřit dopočtením hodnoty kritéria (tedy změnit typ z „P“ na „C“), či je-li možné odříznutí rovnou provést (typ „C“). Informace z typových vektorů je dále využíváno při učení predikčního mechanismu. Záznamy ve *vektoru příspěvků* ***A*** a *vektoru čítačů* ***B*** jsou aktualizovány pouze v případě, že hodnota kritéria před odebráním i po odebrání příznaku je typu „C“. Pokud bychom k aktualizaci predikční informace použili rozdílu predikovaných či smíšených hodnot kritéria, došlo by k zanesení chybné informace a funkčnost predikčního mechanismu by byla vážně narušena.

Obrázek 13.4 ilustruje popsaný mechanismus. Ukazuje průběh učení predikčního mechanismu, kdykoliv dojde ke skutečnému vyhodnocení dvou následných hodnot kritéria včetně následného využití naučené informace k predikci v podobě jednoduchého

odečtení. Predikované hodnoty jakožto pouhé aproximace hodnot skutečných nemohou vést k odříznutí podstromu a každá zdánlivá možnost odřezu musí být proto ověřena spočtením hodnoty skutečné (uzly reprezentující množiny 1, 3, 6 a 1, 2, 6).

Výkonnost predikčního mechanismu se v průběhu algoritmu obvykle postupně zlepšuje. Na počátku, a to vzhledem k nedostatku informace, můžeme očekávat časté aktualizace vektoru příspěvků oproti ojedinělým predikcím. S postupem času a přibýváním „věrohodných" odhadů algoritmus využívá stále častěji predikce na úkor skutečných výpočtů, čímž naopak zpomaluje proces učení predikčního mechanismu. (V této souvislosti byl zaveden nepovinný parametr δ dovolující určit *minimálně nutný počet vyhodnocení* příspěvků příznaku, do jehož dosažení není predikce používána.)



Obr. 13.4 Ilustrace běhu predikčního mechanismu „rychlého algoritmu větví a mezí" na umělém problému výběru $d = 2$ z celkového počtu $D = 6$ příznaků.

Úkolem uvedeného predikčního mechanismu není pouze nahrazovat při průchodu výpočetním stromem výpočty predikcemi, ale také odhadnout okamžik (uzel), kdy již skutečná hodnota kritéria padla pod dosavadní mez a bylo by tedy možné zbytek větve odříznout. Především v tomto smyslu je třeba počítat s možností dvou typů chyb vyplývajících z principu predikce. Nazvěme *pesimistickou chybou predikce* situaci, když FBB zastaví predikci příliš brzy a začne místo ní vyhodnocovat skutečné hodnoty kritéria v důsledku mylného odhadu, že již lze podstrom odříznout. Označme naopak *optimistickou chybou predikce* situaci, pokud při průchodu stromem predikované hodnoty nenaznačují možnost odříznutí podstromu a algoritmus tedy pokračuje zbytečně průchodem do nižších úrovní. Chování algoritmu lze ovlivnit zavedením dodatečné *konstanty optimismu* γ takto: kdykoliv má dojít k predikci odečtením hodnoty zaznamenané ve vektoru příspěvků $\boldsymbol{A}$, nechť je odečtena tato hodnota vynásobená veličinou γ. Hodnoty $\gamma > 1$ takto způsobí rychlejší pokles hodnot kritéria a tím rychlejší přiblížení k dosavadní mezi, a tedy působí proti výskytu *optimistických* chyb predikce ve prospěch chyb *pesimistických*. Hodnoty $0 < \gamma < 1$ působí opačně, více viz odst. 13.4.5.1.

Pro formální popis rychlého algoritmu větví a mezí (Fast Branch & Bound FBB) použijeme značení zavedené při popisu algoritmu BBPP (odst. 13.4.3) rozšířené o tyto konstanty a symboly:

$\delta \geq 1$ – *minimální nutný počet vyhodnocení* (standardně 1),

$\gamma \geq 0$ – *konstanta optimismu* (standardně 1),

$\boldsymbol{T}_k = [T_{k,1}, T_{k,2},\ldots, T_{k,q_k}]^{\mathrm{T}}$, $T_{k,i} \in \{\mathsf{C}\ \mathsf{P}\}$ pro $i = 1,\ldots, q_k$ – *typový vektor*,

(zaznamenává způsob získání hodnot $J_{k,i}$)

$\boldsymbol{V} = [v_1, v_2,\ldots, v_{q_k}]^{\mathrm{T}}$ – *pomocný řadicí vektor*.

Poznámka: Hodnoty q_j množiny Q_j a vektory $\boldsymbol{J}_j$, $\boldsymbol{T}_j$ musí být vždy uloženy pro všechna $j = 0,\ldots, k$, aby byl umožněn zpětný průchod stromem (backtracking).

Kdykoliv dojde v průběhu výpočtu v souvislosti s odebráním nějakého příznaku y_i z nějaké aktuální „kandidátské“ podmnožiny $\overline{\chi}_k$ v k-té úrovni stromu k výpočtu hodnoty kritéria $J(\overline{\chi}_k \setminus \{y_i\})$ a zároveň byla předchozí hodnota $J(\overline{\chi}_k) \equiv J(\overline{\chi}_{k-1} \setminus \{y_i\})$ (po předchozím odebrání nějakého příznaku y_j) taktéž spočítána (což je vyjádřeno $T_{k-1,y_i} = \mathsf{C}$), bude následovat aktualizace predikční informace

$$A_{y_i} = \frac{A_{y_i} \cdot B_{y_i} + J_{k-1,y_i} - J(\overline{\chi}_k \setminus \{y_i\})}{B_{y_i} + 1} \tag{13.5}$$

$$B_{y_i} = B_{y_i} + 1. \tag{13.6}$$

Rychlý algoritmus větví a mezí (FBB)

Stanovme inicializaci stejně jako v případě algoritmu BBPP popsaného v odst. 13.4.3 a navíc nastavme hodnoty δ a γ podle doporučení podaných v odst. 13.4.5.1.

KROK 1: *Zvolme následníky právě zpracovávaného uzlu, a tím vytvořme další úroveň stromu*. Nejprve určeme jejich počet $q_K = r - (D - d - k - 1)$. Nyní sestavme uspořádanou množinu Q_k a vektory $\boldsymbol{J}_k$ a $\boldsymbol{T}_k$ takto: pro všechny příznaky $\psi_j \in \Psi, j = 1,..., r$, je-li $k + 1 < D - d$ (uzly nejsou listy), a $B_{\psi_j} > \delta$ (predikce povolena) položme

$$v_j = J_{k-1,q_{k-1}} - A_{\psi_j},$$

neboli předpovídáme odečtením predikční hodnoty odpovídající příznaku ψ_j od hodnoty kritéria získané v rodičovském uzlu. V jiných případech musí být hodnota kritéria spočtena

$$v_j = J(\overline{\chi}_k \setminus \{\psi_j\}).$$

Po získání všech hodnot v_j tyto hodnoty setřiďme vzestupně

$$v_{j_1} \le v_{j_2} \le \ldots \le v_{j_r}$$

a pro $i = 1,..., q_k$ položme

$Q_{k,i} = \psi_{j_i}$,

$J_{k,i} = v_{j_i}$, pokud v_{j_i} je spočítaná hodnota, nebo

$J_{k,i} = J_{k-1,q_{k-1}} - \gamma \cdot A_{\psi_{j_i}}$ v případě, že v_{j_i} je hodnota predikovaná,

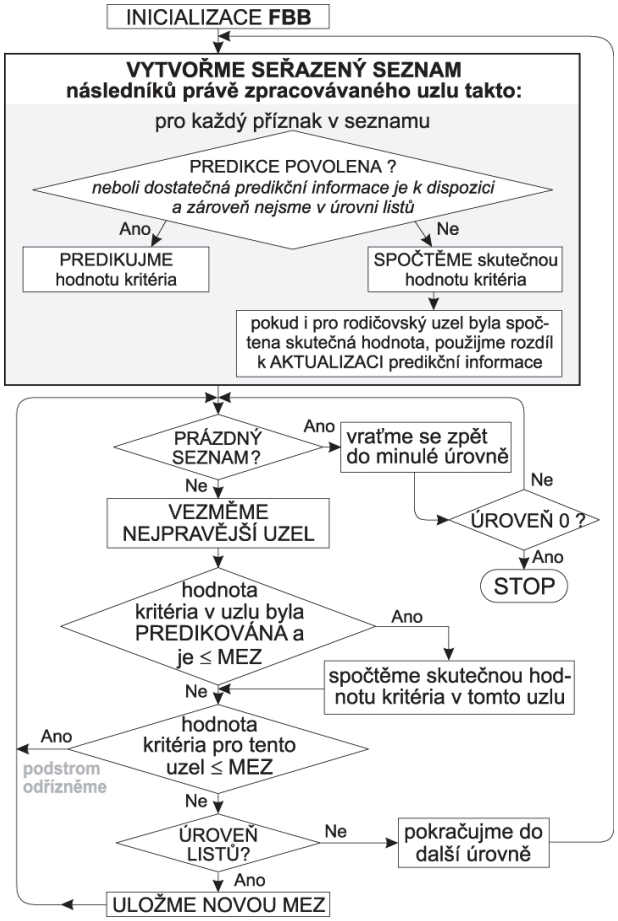
$T_{k\,i}$ = C, pokud v_{j_i} je spočítaná hodnota, nebo

$T_{k,i}$ = P v případě, že v_{j_i} je hodnota predikovaná.

Položme $\Psi = \Psi \setminus Q_k$ a $r = r - q_k$ pro zamezení redundantního vyhodnocování kritéria.

KROK 2: *Otestujme nejpravějšího následníka (připojeného hranou Q_{k,q_k})*. Pokud q_k = 0, potom všichni následníci již byli otestováni, a proto pokračujme Krokem 4 (backtracking). Pokud $T_{k,q_k} = \mathsf{P}$ a $J_{k,q_k} < X^*$, spočtěme skutečnou hodnotu $J_{k,q_k} = J(\overline{\chi}_k \setminus \{Q_{k,q_k}\})$ a označme $T_{k,q_k} = \mathsf{C}$. Pokud $T_{k,q_k} = \mathsf{C}$ a $J_{k,q_k} < X^*$, pokračujme Krokem 3. Jinak položme $\overline{\chi}_{k+1} = \overline{\chi}_k \setminus \{Q_{k,q_k}\}$. Pokud $k + 1 = D - d$, pak byl dosažen list a proto pokračujme Krokem 5. Jinak pokračujme průchodem do další úrovně stromu a položme $k = k + 1$a jděme na Krok 1.

KROKY 3 až 5 zůstávají stejné jako v algoritmu BBPP.



Obr. 13.5 Zjednodušený diagram tzv. rychlého algoritmu větví a mezí (FBB).

Poznámka: V Kroku 1 pro $k = 0$ výraz $J_{-1,q_{-1}}$ označuje hodnotu kritéria na množině všech příznaků $J(X_D)$. Zjednodušený diagram algoritmu naleznete na obr. 13.5.

13.4.5 Nové vlastnosti algoritmů využívajících predikce

Výkonnost kteréhokoliv algoritmu větví a mezí podstatně závisí na vlastnostech použité kriteriální funkce a zpracovávaných dat. Mimo tyto externí faktory, které ovlivňují vyhledávací proces ve všech variantách algoritmu (budou zkoumány podrobněji v odst. 13.4.6), vykazují nové predikční algoritmy BBPP a FBB několik specifických vlastností. Oba algoritmy využívají predikčního mechanismu založeného na heuristických předpokladech. Z toho důvodu nelze spoléhat na bezvadnou funkčnost prediktoru za všech okolností. Algoritmy BBPP i FBB budou nejefektivnější a nejvýrazněji překonají IBB, pokud jejich prediktor uspěje jak ve fázi učení, tak v pozdější fázi náhrady skutečných kriteriálních hodnot hodnověrnými odhady. Toto lze očekávat, zejména pokud individuální příspěvky příznaku k celkové hodnotě kritéria nebudou silně kolísat v kontextu různých podmnožin. V odstavci 13.4.6 ukazujeme, že tento předpoklad není příliš omezující alespoň v kontextu výběru příznaků.

Za předpokladu, že nedojde k naprostému výpadku predikčního mechanismu, bude algoritmus FBB znatelně rychlejší než BBPP. Algoritmus FBB obsahuje totiž nejen všechna vylepšení obsažená v BBPP, ale přidává další. Na druhou stranu výkonnost BBPP méně závisí na úspěšnosti predikce. Potenciální selhání predikce by v případě BBPP mělo pouze nepřímý vliv na výslednou rychlost, špatné řazení vnitřních uzlů stromu omezí efektivitu odříznutí podstromů, avšak zůstává základní výhoda oproti IBB v podobě vždy nižšího počtu výpočtů prováděných při konstrukci jednotlivých vnitřních úrovní stromu.

Oproti klasickým algoritmům větví a mezí oba algoritmy FBB i BBPP spotřebovávají určitý čas navíc správou predikčního mechanismu. Tento čas je ovšem zanedbatelný ve srovnání s časem ušetřeným vynecháním vyhodnocování kritéria zejména v případě nerekurzivních forem kritérií v kontextu výběru příznaků. BBPP je na rozdíl od FBB použitelný i s rekurzivními formami kritéria (viz obr. 13.7).

Poznamenejme, že kromě urychlení na základě predikce je aktuálně zkoumána i možnost prodeje prostoru za čas (Chen, 2003). Za cenu budování rozsáhlé paměťové struktury je možné odhalit některá další vyhodnocování kritéria jako zbytečná. V tomto přístupu je zatím ale obtížné udržet časově správu pomocných struktur v rozumných mezích.

13.4.5.1 Specifické vlastnosti rychlého algoritmu větví a mezí

Algoritmus FBB nemůže být použit s rekurzivními formami kritérií, v nichž je k výpočtu hodnoty $J(\overline{\chi}_k)$ třeba znát přesně předchozí hodnotu $J(\overline{\chi}_{k-1})$. Algoritmus FBB při průchodu větvemi nahrazuje výpočty hodnot kritéria predikovanými odhady, jejichž využití k rekurzivnímu výpočtu hodnot následujících by zatížilo systém neúnosnou chybou. Výhoda algoritmu FBB je tím výraznější, čím větší je výpočetní náročnost použité nerekurzivní formy kritéria.

Konstanty γ a δ mohou být volitelně použity k úpravě funkčnosti predikčního mechanismu. Za základní a v obecném případě nejvýhodnější nechť jsou považovány hodnoty $\gamma = 1$ a $\delta = 1$. Pro konkrétní problémy je možné že odlišné hodnoty by výpočet

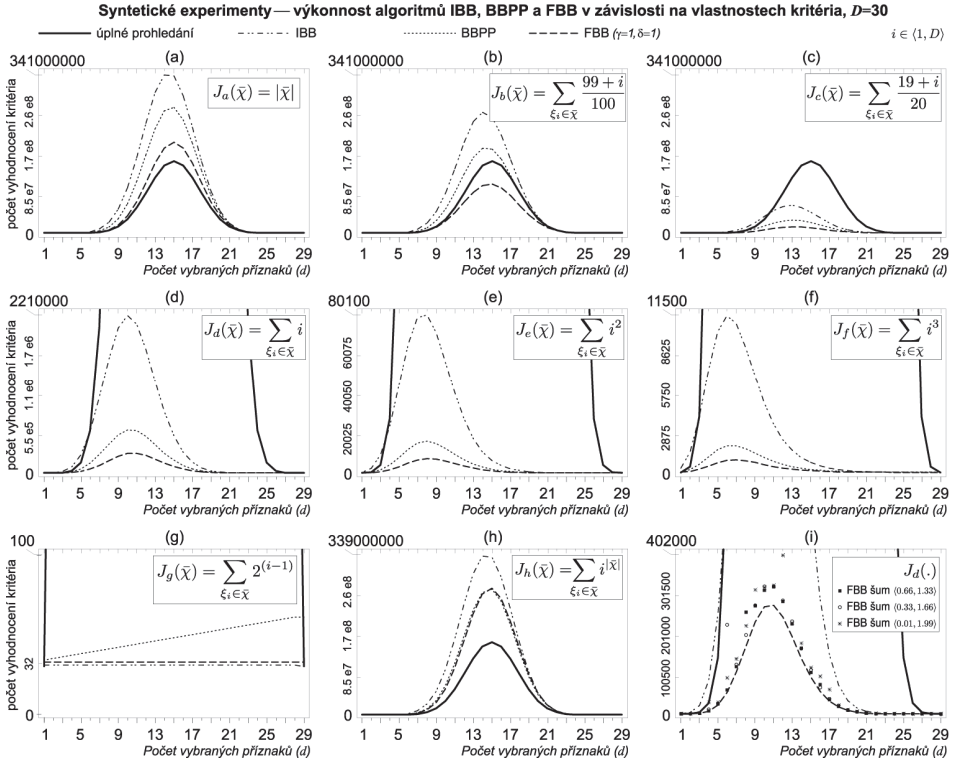
urychlily. Bohužel není k dispozici žádný způsob, jak konkrétní optimální hodnoty odhadnout jinak než na základě proběhlého výpočtu. Přesto je možné říci že konstanta optimismu γ ovlivňuje běh algoritmu FBB významněji než konstanta minimálního počtu vyhodnocení δ. Hodnoty $\gamma > 1$ způsobují pesimistické chování algoritmu, neboli mohou sloužit k omezení výskytu potenciálně nebezpečnějších optimistických chyb predikce. Čím pesimističtější je nastavení algoritmu, tím méně je prováděno predikcí a algoritmus se více přibližuje klasickému IBB (popř. BBPP). Přesněji řečeno predikce budou využívány pouze ve vyšších úrovních stromu a budou zastavovány předčasně při průchodu stromem směrem k větvím. Hodnoty $0 < \gamma < 1$ naopak způsobují méně žádoucí optimistické chování algoritmu, které může vyústit ve zpomalení až pod úroveň IBB. Pro $\gamma = 0$ se pak algoritmus stává ekvivalentem úplného prohledání.

13.4.6 Experimenty s optimálním výběrem příznaků

Pro ilustraci chování uváděných algoritmů jsme připravili mimo experimenty s reálnými daty a standardními pravděpodobnostními kritérii též různé umělé testy nezávislé na datech, které umožňují demonstrovat změny výkonnosti algoritmů v různých modelových situacích. Zdůrazněme, že všechny uvažované algoritmy jsou optimální vzhledem ke zvolené kriteriální funkci a ve výsledku nacházejí tutéž podmnožinu příznaků. Výsledná klasifikační výkonnost je tedy ve všech případech stejná a závisí na faktorech zde nezkoumaných (volbě kritéria apod.). Zde se soustředíme na jediný významný rozdíl mezi různými optimálními algoritmy, totiž na rozdíl v rychlosti výpočtu.

13.4.6.1 Syntetické experimenty

Výkonnost kteréhokoliv algoritmu větví a mezí podstatně závisí na vlastnostech použitého kritéria a datového souboru. Abychom mohli ukázat vlastnosti algoritmů co nejobecněji, nadefinovali jsme sadu ilustrativních kriteriálních funkcí nezávislých na datech. Obrázek 13.6(a) ukazuje nejhorší možnou situaci, kdy je všem variantám algoritmu větví a mezí znemožněno odřezávat podstromy, tato situace je vyvolána prostřednictvím kriteriální funkce ohodnocující všechny příznaky jako ekvivalentní. Následující série obrázků (b) až (f) ukazuje jak rostoucí vzájemná odlišnost příznaků (ve smyslu jejich příspěvku k hodnotě kritéria) vede k efektivnějšímu odřezávání podstromů. Je vidět, že algoritmy založené na predikci dokáží této rostoucí rozmanitosti příznaků využít lépe než klasický algoritmus IBB. Extrémní případ vidíme na obr. 13.6(g), kde každý příznak přispívá k hodnotě kritéria větší hodnotou, než je součet příspěvků všech příznaků s menším indexem. V důsledku dokáží všechny varianty algoritmu s *horizontálním řazením uzlů* konstruovat výpočetní strom, v němž je optimum nalezeno již při průchodu první větví a všechny zbylé větve jsou ihned poté odříznuty. V tomto případě dojde k nalezení optima v lineárním čase. Obrázek 13.6(h) ukazuje, jakým způsobem lze vyřadit z činnosti predikční mechanismus algoritmu FBB. Takováto situace nastává, je-li hodnota kritéria silně závislá na velikosti zkoumané podmnožiny. Na obrázku 13.6(i) vidíme klesající efektivitu FBB v důsledku vlivu šumu na učení – zde násobíme v průběhu učení každý zaznamenávaný skutečný příspěvek příznaku náhodným koeficientem generovaným podle rovnoměrného rozložení z vhodného intervalu. Všimněme si, že ani poměrně výrazné zašumění (interval [0,01 1,99]) nesníží celkovou efektivitu algoritmu FBB na úroveň IBB.



Obr. 13.6 Umělá kritéria demonstrující chování algoritmu větví a mezí. (a) Nejhorší možný scénář zabraňující zcela odříznutím podstromů. (b) až (f) Série příkladů ukazující efekt zlepšování výkonnosti se vzrůstajícími rozdíly mezi příznaky. (g) Nejlepší možný scénář vedoucí k nejefektivnějšímu možnému odřezávání podstromů. (h) Scénář selhání učení v algoritmu FBB. (i) Vliv šumu na učení v algoritmu FBB.

Z předchozích pozorování lze vyvodit, že výkonnost algoritmů větví a mezí závisí především na následujících charakteristikách procesu vyhodnocování kritéria:

1. Všechny příznaky by mezi sebou měly vykazovat stabilní a měřitelné odlišnosti ve smyslu příspěvku k hodnotě kritéria vyhodnocovaného na libovolných podmnožinách. Čím podobnější jsou příspěvky jednotlivých příznaků, tím horší výkonnost kteréhokoliv algoritmu větví a mezí lze očekávat. Příčina je prostá – prochází-li algoritmus od kořene k listům, aktuální hodnota kritéria klesá o příspěvky jednotlivých odebíraných příznaků. V případě velmi podobných příspěvků u všech příznaků bude aktuální hodnota kritéria velmi podobná v na libovolném uzlu na stejné úrovni stromu. Je tedy méně pravděpodobné, že v některém uzlu dané úrovně časem klesne pod mez a umožní tak odříznutí podstromu.

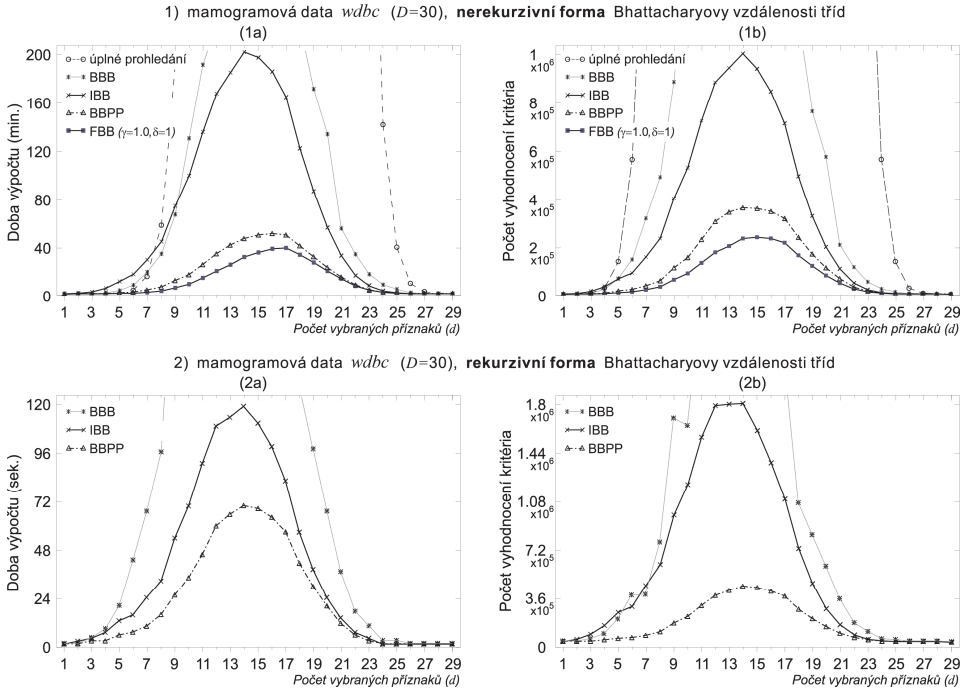
2. Příspěvek příznaku by neměl příliš kolísat v kontextu různých podmnožin. Nestabilní či zašuměné příspěvky příznaku zhoršují učící schopnost predikčního mechanismu (s důsledky hlavně v FBB). Navíc heuristika horizontálního řazení v takovém případě produkuje více či méně náhodná pořadí uzlů stromu, a tím podstatně sníží efektivitu odřezávání podstromů ve všech algoritmech IBB BBPP a FBB (viz obr. 13.6(i)).

3. Hodnota kritéria by neměla záviset příliš silně na velikosti vyhodnocované podmnožiny (neboli na momentální úrovni stromu). Příliš vysoké rozdíly mezi příspěvkem téhož příznaku ve vztahu k různě velkým podmnožinám mohou opět zhoršit funkčnost predikčního mechanismu v FBB. Heuristika horizontálního řazení algoritmů IBB BBPP a FBB také v tomto případě produkuje nevýhodná pořadí uzlů (viz obr. 13.6(h)).

Celkově lze pozorovat, že algoritmy využívající predikce (BBPP FBB) vykazují větší efektivitu než referenční IBB pro všechna testovaná syntetická kritéria, ať už šlo o kritéria uměle komplikující či urychlující hledací proces.

13.4.6.2 Experimenty na reálných datech

Algoritmy zde testujeme na mamogramových datech *wdbc* o dvou třídách z Wisconsin Diagnostic Breast Centra (30 příznaků, 357 záznamů o zdravých a 212 nemocných pacientech) a datech *waveform* o třech třídách (40 příznaků, 1691 vzorků třídy 1. třídy a 1693 vzorků třídy 2., třetí třída je vynechána vzhledem k omezení kriteriální funkce). Oba datové soubory pocházejí z databáze UCI (Asuncion a kol. 2007). Dále jsme použili řečová data *speech* o dvou třídách pocházející z British Telecomu (15 příznaků, 682 záznamů slova „yes“ a 736 záznamů slova „no“) získaná z CVSSP University of Surrey. Abychom zdůraznili univerzálnost zkoumaných algoritmů, provedli jsme testy se třemi různými kriteriálními funkcemi: divergencí, Bhattacharyovou vzdáleností a Patrickovou–Fischerovou vzdáleností (Devijver a kol., 1982). Použili jsme jak rekurzivní (bylo-li

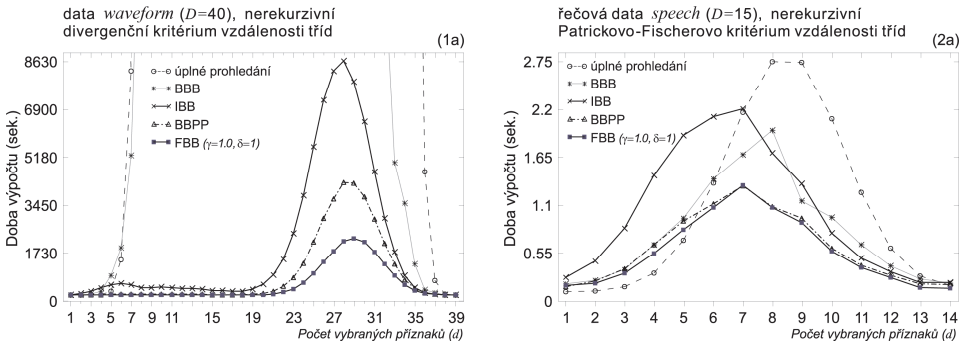


Obr. 13.7. Výkonnost optimálních metod pro hledání podmnožin na mamogramových datech wdbc o 2 třídách.

to možné), tak nerekurzivní formu Bhattacharyovy vzdálenosti. Výkonnost různých metod je ilustrována na obrázcích 13.7 a 13.8, které obsahují (a) graf výpočetního času a v případě dat *wdbc* též (b) graf počtu skutečně vyhodnocených hodnot kritéria. Tabulka 13.2 ukazuje statistiky související s učením predikčního mechanismu v algoritmech BBPP a FBB (k výpočtu statistik byl použit algoritmus IBB, který vyhodnocuje více hodnot a poskytuje tak informativnější výsledky). Tabulka ukazuje vektory ***A*** (průměrné příspěvky příznaků), ***B*** (počty vyhodnocení příspěvku příznaků), odchylku vyhodnocovaných hodnot A_i a individuální hodnoty kritéria $J(\{i\})$ spočtené pro jednotlivé příznaky zvlášť. Při použití nerekurzivních kriteriálních forem jsme nastavili všechny algoritmy tak, aby konstruovaly tzv. strom minimálního řešení (Yu a kol., 1993), což není možné v případě forem rekurzivních. Parametry FBB byly nastaveny na hodnoty $\gamma = 1$ a $\delta = 1$.

Tabulka 13.2 Statistiky příspěvků příznaků k hodnotě kritéria pro datový soubor *wdbc*. Spočteno algoritmem IBB při výběru 15 z 30 příznaků za použití Bhattacharyovy vzdálenosti. Nalezená podmnožina: $X = \{1, 3, 4, 6, 7, 11, 14, 15, 16, 17, 21, 23, 24, 26, 27\}$; $J_B(X) = 5{,}741$; $J_B(X_D) = 7{,}53989$

i	1	2	3	4	5	6	7	8	9	10	11
A_i	0,269	0,066	0,272	0,651	0,118	0,114	0,195	0,130	0,085	0,077	0,214
A_i *st. odch.*	0,076	0,022	0,098	0,048	0,030	0,020	0,041	0,029	0,028	0,019	0,087
B_i	312	95630	525	4	30327	56149	1400	23549	93275	89565	395
$J_B(\{i\})$	0,586	0,113	0,629	0,641	0,073	0,307	0,498	0,790	0,064	0,003	0,463
i	12	13	14	15	16	17	18	19	20	21	22
A_i	0,119	0,126	0,843	0,127	0,133	0,441	0,122	0,105	0,198	0,340	0,083
A_i *st. odch.*	0,021	0,040	0	0,034	0,031	0,004	0,026	0,015	0,051	0,108	0,026
B_i	51923	12972	1	10083	7746	7	31892	77102	452	31	91836
$J_B(\{i\})$	0,010	0,483	0,803	0,003	0,050	0,081	0,103	0,032	0,036	0,787	0,140
i	23	24	25	26	27	28	29	30			
A_i	0,267	0,797	0,095	0,199	0,103	0,132	0,123	0,094			
A_i *dev.*	0,084	0,011	0,017	0,031	0,027	0,019	0,015	0,032			
B_i	142	2	88359	1037	9441	32543	51707	38630			
$J_B(\{i\})$	0,816	0,811	0,110	0,327	0,389	0,820	0,175	0,101			



Obr. 13.8 Výkonnost optimálních metod pro hledání podmnožin na datech *waveform* a *speech* o dvou třídách.

Exponenciální charakter problému optimálního vyhledávání je dobře ilustrován faktem, že pro 15dimenzionální data je ještě úplné prohledání srovnatelně rychlé s algo-

ritmy větví a mezí, zatímco pro data 30dimenzionální při hledání podmnožiny $d = 15$ (obrázek 13.7) již úplné prohledání potřebuje vyhodnotit přibližně 140krát více hodnot kritéria než referenční algoritmus větví a mezí IBB. U 40dimenzionálních dat při hledání podmnožiny $d = 20$ (obrázek 13.8) je tento rozdíl již $1{,}3.10^8$násobný.

Podobně jako v případě předchozích umělých testů všechny výsledky (obrázky 13.7 a 13.8) potvrzují, že oba algoritmy využívající predikce (BBPP a FBB) nacházejí řešení nejrychleji, v podstatě vždy překonávají referenční IBB. V závislosti na datovém souboru a použitém kritériu algoritmus FBB prokazuje schopnost řešit úlohu výběru příznaků 1,5× až 10× rychleji než IBB, zejména při výpočetně nejnáročnější volbě d blízko $D/2$.

Příklad *wdbc* (obrázek 13.7) ukazuje očekávatelné chování zkoumaných algoritmů, v tomto případě s využitím Bhattacharyovy vzdálenosti jakožto kritéria. Tabulka 13.2 v tomto případě popisuje velmi přesně naučenou predikční informaci (jak potvrzuje nízká odchylka A_i). Rozdíly mezi jednotlivými příznaky jsou viditelné a stabilní. V souladu s tím heuristika horizontálního řazení i predikční mechanismy úspěšně splnily svou funkci.

Příklad *waveform* (obrázek 13.8) ukazuje zajímavý fenomén, kdy všechny algoritmy s horizontálním řazením IBB BBPP a FBB běžely překvapivě rychle až do velikosti hledané podmnožiny $d = 20$. Nabízí se vysvětlení, že zhruba polovina příznaků tohoto datového souboru reprezentuje pouze neinformativní šum (potvrzeno dokumentací souboru). Rozdíl mezi odhadnutými příspěvky informativních a šumových příznaků zde byl výborně využit heuristikou horizontálního řazení uzlů stromu.

Příklad řečových dat *speech* (obrázek 13.8) představuje úlohu, při níž pokročilé mechanismy modernějších algoritmů větví a mezí částečně selhávají především kvůli vlastnostem použitého Patrickova–Fischerova kritéria, které je silně závislé na velikosti ohodnocované podmnožiny. Tento příklad také ukazuje, že úplné prohledání může být nejvýhodnější volbou zejména pro hodnoty d velmi blízké 0, u nichž může průchod hlubokého výpočetního stromu vzhledem k relativně malému počtu listových konfigurací algoritmy větví a mezí rozhodujícím způsobem znevýhodnit.

Tabulka 13.2 ilustruje nejen příspěvky jednotlivých příznaků, ale také efektivitu heuristiky horizontálního řazení v algoritmech IBB BBPP a FBB. Všimněme si, že příznaky s nízkým příspěvkem k hodnotě kritéria jsou vyhodnocovány častěji než příznaky s příspěvkem vysokým. Horizontální řazení takto podporuje co nejrychlejší růst meze. Významné příznaky s vysokým příspěvkem jsou akceptovány rychleji, a tím vynechány z dalšího prohledávání. Tabulka též ilustruje známý fakt, že individuální hodnoty kritéria spočítané zvlášť pro jednotlivé příznaky nevypovídají dostatečně o skutečném významu příznaku. Všimněme si, jak výrazně se některé hodnoty $J(\{i\})$ liší od průměrných příspěvků příznaků A_i.

13.4.7 Shrnutí optimálních metod

Jedinou metodou hledání optimální podmnožiny použitelnou s nemonotónními kritérii (tedy je-li například kritériem odhad chyby klasifikace) je *úplné prohledání*. Vzhledem k exponenciálnímu charakteru problému optimálního výběru příznaků (a tedy extrémní výpočetní náročnosti) je již pro úlohy relativně malé dimenzionality nutné hledat alternativní metody i za cenu omezení se na monotónní kritéria. Řada nedávných vylepšení *algoritmu větví a mezí* především v podobě predikčních algoritmů FBB a BBPP vyústila

do deseti- až stonásobného zrychlení oproti základní variantě algoritmu (v závislosti na konkrétním datovém souboru a kritériu).

Je ovšem potřeba zdůraznit, že i přes uvedený pokrok zůstávají optimální metody v principu exponenciální a tedy velmi pomalé. Není-li pro jejich aplikaci zvláštní důvod, bývá praktičtější využít některou z metod suboptimálních. Suboptimální metody jsou obvykle flexibilnější a schopné řešit problémy podstatně větších dimenzionalit, přičemž nalézaná řešení nemusí být v konečném důsledku nutně horší než řešení optimální. Volba vhodného kritéria má větší význam než časem draze zaplacené nalezení optima vzhledem k nevhodně zvolenému kritériu. Pro podrobnější diskusi optimálních metod odkazujeme na článek (Somol a kol., 2004).

13.5 Suboptimální vyhledávací metody

Přes pokroky v oblasti optimálních metod výběru příznaků (Somol a kol., 2004), (Nakariyakul a kol., 2007) je u většiny reálných úloh v kontextu statistického rozpoznávání obvykle nutné se obracet k suboptimálním metodám.

Známe velké množství nejrůznějších suboptimálních metod založených na nejrůznějších principech a poskytujících různou míru kompromisu mezi efektivitou a kvalitou nalezeného řešení. Velké množství metod je založeno na sledování největšího gradientu v prostoru přípustných řešení – nejtypičtějšími metodami jsou metody sekvenčního vyhledávání (Devijver a kol., 1982), metody Plus-*l*-Minus-*r* (Devijver a kol., 1982), jejich zobecněné verze (Devijver a kol., 1982), plovoucí vyhledávání (Pudil a kol., 1994a), (Somol a kol., 1999), (Nakariyakul a kol., 2009), oscilační vyhledávání (Somol a kol., 2000), (Somol a kol., 2008b) atd. Deterministické metody však mohou být náchylné k uvíznutí v lokálním extrému, proto je velmi často v nějaké formě využívána randomizace vyhledávacího procesu – hlavními zástupci těchto přístupů jsou metody Las Vegas, metoda Relief (Kononenko, 1994), (Sun, 2007), metoda Focus (Arauzo-Azofra a kol., 2003), simulované žíhání, např. (Debuse a kol., 1997), evoluční (genetické) algoritmy (Hussein a kol., 2001), vyhledávání tabu (Zhang a kol., 2002), metody mravenčí kolonie, např. (Jensen, 2006), randomizované oscilační vyhledávání (Somol a kol., 2000), ale také metoda řazení respektující závislost příznaku DAF (Somol a kol., 2011).

Pro výpočetně nejnáročnější problémy velmi velké dimenzionalty jsou obvykle zanedbávány vztahy mezi příznaky (metoda nejlepších individuálních příznaků, Best Individual Features, BIF, viz např. (Yang a kol., 1997)), popř. jsou přijímána nejrůznější zjednodušení či omezení rozsahu prohledávání. Řada metod umožňuje omezit prostor prohledávání uživatelskými parametry (Pudil a kol., 1994a) (Somol a kol., 2000) a (Somol a kol., 2008b), popř. se předpokládá, že uživatel pomocí parametrů rozsah prohledávání přesně specifikuje (Hussein a kol., 2001) a (Zhang a kol., 2002).

Ještě významnější než volba metody a jejích parametrů se ukazuje volba kriteriální funkce (viz filtry vs. pouzdra, odst. 13.3.2). Při nevhodné volbě kriteriální funkce nelze ani v případě nalezení optima očekávat optimální přesnost výsledného klasifikátoru. Tomuto a dalším problémům ovlivňujícím výsledek výběru příznaků se věnujeme též v oddílu 13.9.

Integrální součástí každého procesu výběru příznaků je rozhodnutí o počtu příznaků, které máme vybrat. Určení správné dimenzionality podprostoru vybraných příznaků je

obtížný problém, který je v obecnosti nad rámec této kapitoly. V další diskusi budeme nicméně rozlišovat dva typy metod FS: tzv. metody *d*-parametrizované a *d*-optimalizující. Většina existujících metod je *d*-parametrizovaných, tj. vyžadujících od uživatele rozhodnutí o kardinalitě výsledné podmnožiny příznaků. V oddílu 13.6 bude popsána *d*-optimalizující deterministická procedura, která optimalizuje současně jak velikost podmnožiny, tak její složení. Mezi nedeterministické *d*-optimalizující metody patří zejména evoluční algoritmy (Caruana a kol., 1994), (Chaikla a kol., 1999), (Mayer a kol., 2000) a (Hussein a kol., 2001).

V následujícím textu uvádíme základní přehled několika hlavních nástrojů užitečných pro problémy různé složitosti, založených většinou na konceptu sekvenčního vyhledávání (viz odst. 13.5.2).

13.5.1 Individuálně nejlepší příznaky

Metoda *individuálně nejlepších příznaků* (Best Individual Features, BIF) je nejrychlejší metodou výběru příznaků. Každý příznak je nejprve individuálně ohodnocen za použití zvoleného kritéria. Podmnožiny jsou pak vybrány prostě volbou prvních *d* individuálně nejlepších příznaků. Tento přístup je nejjednodušší, ale zároveň potenciálně nejslabší z hlediska optimalizační schopnosti – při výběru totiž zcela ignoruje možné komplikované závislosti mezi příznaky. Pro řešení problému FS při velké počáteční dimenzionalitě je to však často jediný aplikovatelný přístup. Metoda BIF je takto standardem pro kategorizaci textových dokumentů (Yang a kol., 1997), (Sebastiani, 2002), vyhledávání genů (Xing, 2003) (Saeys a kol., 2007) a v dalších aplikacích, kde dimenzionalita problému roste do tisíců.

Metoda BIF může být ovšem výhodná i v jiných situacích, protože se uplatňuje úspěšně tam, kde pokročilejší metody selhávají a produkují nestabilní výsledky, popř. vedou k přeučení a ztrátě schopnosti zobecnění (viz oddíl 13.9). Tyto situace nastávají nejen při problémech velmi velké dimenzionality, ale složitější metody jsou obecně tím náchylnější k selhání, čím větší je nepoměr mezi příliš velkou dimenzionalitou vůči příliš malému počtu vzorků trénovacího souboru. Není-li však úloha zatížena zmíněnými omezujícími okolnostmi, pokročilejší metody výběru příznaků mají větší potenciál dosáhnout lepších výsledků.

13.5.2 Sekvenční vyhledávání

Pro zjednodušení další diskuse se zaměříme na sekvenční vyhledávací metody. Označme Y množinu všech příznaků ($|Y| = D$). Většina známých sekvenčních algoritmů výběru příznaků sdílí stejný základní mechanismus přidávání a odebírání příznaků do stávající pracovní podmnožiny a z ní pryč. Příslušné kroky lze popsat uvedenými definicemi (pro zjednodušení budeme uvažovat pouze nezobecněné algoritmy (Devijver a kol., 1982), které zpracovávají v daném okamžiku pouze jeden příznak).

Definice 1. Pro stávající množinu příznaků X_d nechť f^+ je příznak takový, že

$$f^+ = \arg\max_{f \in Y \setminus X_d} J^+(X_d, f), \tag{13.7}$$

kde $J^+(X_d, f)$ označuje kriteriální funkci použitou pro vyhodnocení podmnožiny získané přidáním f do X_d, kde $f \in Y \setminus X_d$. Pak budeme říkat, že $ADD(X_d)$ je operace přidání příznaku f^+ do X_d za účelem vytvoření množiny X_{d+1}, jestliže

$$ADD(X_d) \equiv X_d \cup \{f^+\} = X_{d+1}, X_d, \qquad X_{d+1} \subset Y. \tag{13.8}$$

Definice 2. Nechť f^- je ve stávající množině příznaků X_d takový příznak, že

$$f^- = \arg\max_{f \in X_d} J^-(X_d, f), \tag{13.9}$$

kde $J^-(X_d f)$ označuje kriteriální funkci použitou k vyhodnocení podmnožiny získané vyloučením f z X_d, je-li $f \in X_d$. Pak budeme říkat, že $REM(X_d)$ je operace vyloučení příznaku f^- z množiny X_d za účelem vytvoření množiny X_{d-1}, jestliže

$$REM(X_d) \equiv X_d \setminus \{f^-\} = X_{d-1}, \qquad X_d, X_{d-1} \subset Y. \tag{13.10}$$

S cílem zjednodušit označení pro opakující se použití operací FS zavedeme tato užitečná označení:

$$X_{d+2} = ADD(X_{d+1}) = ADD(ADD(X_d)) = ADD^2(X_d),$$
$$X_{d-2} = REM(REM(X_d)) = REM^2(X_d) \tag{13.11}$$

a obecněji

$$X_{d+\delta} = ADD^\delta(X_d), \qquad X_{d+\delta} = ADD^\delta(X_d). \tag{13.12}$$

Poznamenejme, že ve standardních sekvenčních metodách výběru příznaků $J^+(.)$ a $J^-(.)$ označují

$$J^+(X_d, f) = J(X_d \cup \{f\}),\ J^-(X_d, f) = J(X_d \setminus \{f\}), \tag{13.13}$$

kde $J(.)$ je kriteriální funkce založená obvykle buď na některé vhodné pravděpodobnostní míře, nebo na odhadu úspěšnosti klasifikace (viz filtry a pouzdra, odst. 13.3.2 (Kohavi a kol. 1997)).

13.5.3 Nejjednodušší sekvenční výběr

Nejjednodušší a přitom široce používanou variantou sekvenčního vyhledávání jsou metody *sekvenčního dopředného výběru* (SFS) a *sekvenčního zpětného výběru* (SBS) (Whitney, 1971) (Devijver a kol., 1982). Metoda SFS iterativně přidává (SBS odebírá) vždy v daném kroku jeden příznak tak, aby byla maximalizována okamžitá hodnota kritéria tak dlouho, až je dosaženo požadované velikosti cílové podmnožiny příznaků. Metoda SFS začíná od prázdné množiny a je proto známa jakožto metoda typu *bottom-up*. Metoda SBS začíná od množiny všech příznaků a je proto známa jako metoda typu *top-down*.

Sekvenční dopředný výběr (SFS)

Algoritmus generuje podmnožinu d příznaků zdola:. $X_d = ADD^d(\emptyset)$.

Sekvenční zpětný výběr (SBS)

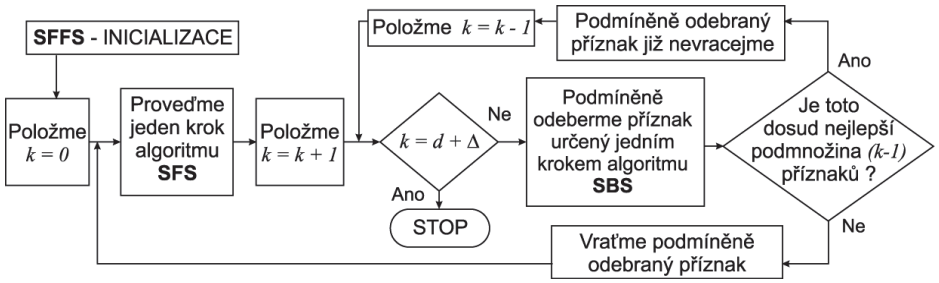
Algoritmus generuje podmnožinu d příznaků shora: $X_d = REM^{|Y|-d}(Y)$.

SBS jakožto metoda typu top-down od počátku výpočtu vyhodnocuje vztahy mezi všemi příznaky. Dává tak teoreticky šanci odhalit složitější vztahy mezi nimi a lépe tak rozhodovat o důležitosti jednotlivých příznaků. Metoda SBS je z téhož důvodu ale také výpočetně náročnější a oproti SFS více náchylná k numerickým problémům. Metoda SFS je z těchto důvodů často upřednostňována.

Jako většina dřívějších sekvenčních metod jak SFS, tak SBS trpí takzvaným hnízděním podmnožin příznaků (rozhodnutí o přidání, popř. odebrání, příznaku není v pozdější fázi výpočtu nikdy revokováno), které významně zhoršuje schopnost dosáhnout optima. Prvním pokusem o překonání tohoto problému bylo použití algoritmu Plus-*l*- Take away-*r* (také známého jako (*l*, *r*)) nebo zobecněného algoritmu (*l*, *r*) (Devijver a kol., 1982), které oba zahrnují postupný proces přidávání i odebírání příznaků. Tatáž základní myšlenka ovšem v podstatně rozšířené a složitější formě představuje základ plovoucího vyhledávání.

13.5.4 Sekvenční plovoucí vyhledávání

Procedura *sekvenčního dopředného plovoucího vyhledávání* (Sequential Forward Floating Search, SFFS) (Pudil a kol., 1994a) spočívá v tom, že po každém dopředném kroku aplikujeme zpětné kroky tak dlouho, dokud jsou výsledné podmnožiny lepší než ty, které byly předtím vyhodnoceny jako nejlepší na dané úrovni (pro danou dimenzionalitu). Z toho plyne, že v případě, kdy přechodný okamžitý výsledek nezlepšuje předchozí výsledek pro danou dimenzionalitu, k žádným zpětným krokům vůbec nedojde. Totéž platí pro dopředné kroky ve zpětné verzi procedury – *sekvenční zpětné plovoucí vyhledávání* (Sequential Backward Floating Search, SBFS). Oba algoritmy dovolují samořízené vratné kroky, které umožňují dosáhnout kvalitního řešení dynamicky flexibilním přepínáním mezi kroky dopřednými a zpětnými. Lze říci, že oba algoritmy provádějí pouze výpočty, které jsou potřeba k vylepšení doposud známých řešení, aniž by bylo nutné nastavovat jakékoliv uživatelské parametry.



Obr. 13.9 Algoritmus sekvenčního dopředného plovoucího výběru.

Sekvenční dopředný plovoucí výběr (SFFS)

Algoritmus generuje podmnožinu d příznaků zdola s volitelným, rozsah prohledávání omezujícím parametrem $\Delta \in [0, D-d]$ (pro hledání v plném rozsahu položme $\Delta = D-d$).

KROK 1: Položme $X_0 = \emptyset$, $k = 0$.

KROK 2: Přidejme k pracovní množině relativně nejlepší příznak pomocí operace $X_{k+1} = ADD(X_k)$, $k = k + 1$.

KROK 3: Opakujme operaci odebrání relativně nejhoršího příznaku pomocí operace $X_{k-1} = REM(X_k)$, $k = k - 1$, tak dlouho, dokud tímto způsobem nacházíme pro menší k lepší než doposud známá řešení.

KROK 4: Pokud $k < d + \Delta$, jděme na Krok 2.

Podrobný formální popis této dnes již klasické procedury lze nalézt v článku (Pudil a kol., 1994a). Nicméně základní myšlenka je poměrně jednoduchá a může být dostatečně ilustrována obrázkem 13.9. (Podmínka $k = d + \Delta$ ukončuje algoritmus poté, co byla nejen dosažena cílová podmnožina d příznaků, ale popřípadě i vylepšena pomocí zpětných vratných kroků z dimenzionalit větších než d.)

Plovoucí vyhledávací algoritmy lze považovat za univerzální nástroje, které nejen překonávají všechny své předchůdce, ale které si přitom uchovávají přednosti, jež nemají pozdější sofistikovanější algoritmy. Mají schopnost nalézat dobrá řešení pro všechny dimenze daného problému jedním průchodem algoritmu. Celková rychlost vyhledávání je dostatečně vysoká pro většinu praktických problémů.

Sekvenční zpětný plovoucí výběr (SBFS)

Algoritmus generuje podmnožinu d příznaků shora s volitelným, rozsah prohledávání omezujícím parametrem $\Delta \in [0, D - 1]$ (pro hledání v plném rozsahu položme $\Delta = D - 1$).

KROK 1: Položme $X_{|Y|} = Y$, $k = |Y|$.

KROK 2: Odeberme z pracovní množiny relativně nejhorší příznak pomocí operace $X_{k-1} = REM(X_k)$, $k = k - 1$.

KROK 3: Opakujme operaci přidání relativně nejlepšího příznaku pomocí operace $X_{k+1} = ADD(X_k)$, $k = k + 1$, tak dlouho, dokud tímto způsobem nacházíme pro větší k lepší než doposud známá řešení.

KROK 4: Pokud $k > d + \Delta$, jděme na Krok 2.

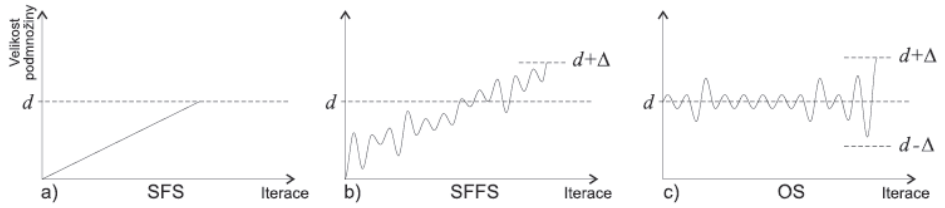
13.5.5 Další rozvoj principu plovoucího vyhledávání

Poté, co se plovoucí algoritmy úspěšně prosadily jako obecně přijímaný univerzální prostředek pro výběr příznaků, byla jejich základní idea dále zkoumána a rozšiřována. V práci (Somol a kol., 1999) byly navrženy tzv. *adaptivní plovoucí algoritmy*, ASFFS a ASBFS. Ty jsou schopny dosáhnout v některých případech lepších výsledků než klasické SFFS a SBFS, ale za cenu podstatného nárůstu času potřebného pro vyhledávání a nutnosti nastavit nepříliš intuitivní uživatelské parametry. Naše zkušenost ukazuje, že algoritmy ASFFS či ASBFS jsou obvykle méně vhodné než novější algoritmy, na které zaměříme svou pozornost v dalším textu. Zdokonalená verze plovoucího vyhledávání byla nedávno publikována v práci (Nakariyakul a kol., 2009).

13.5.6 Oscilační vyhledávání

Novější, tzv. *oscilační vyhledávání* (Oscillating Search, OS) (Somol a kol., 2000) může být považováno za metaproceduru, která využívá jiné metody výběru příznaků jako podprocedury ve svém vlastním vyhledávání. Celý koncept oscilačního vyhledávání je velmi flexibilní a umožňuje modifikace pro různé účely. Ukázal se jako velmi výkonný a schopný překonat klasické sekvenční procedury včetně algoritmů plovoucího vyhledávání.

Na rozdíl od jiných metod je OS založeno na opakované modifikaci stávající podmnožiny příznaků X_d neměnné velikosti d. V tomto smyslu OS nezávisí na převládajícím směru vyhledávání. Toho je dosaženo střídáním tzv. *výkyvů dolů a nahoru*. Výkyvy oběma směry mají za cíl zlepšit stávající podmnožiny X_d nahrazením některých příznaků lepšími. *Výkyv dolů* nejprve odstraňuje a pak vrací zpátky, zatímco *výkyv nahoru* nejprve přidává a poté odstraňuje. Dva po sobě následující výkyvy tvoří *oscilační cyklus*. Na OS lze tedy nahlížet jako na řízenou posloupnost oscilačních cyklů. Hodnota o označovaná jako *hloubka oscilačního cyklu* určuje počet příznaků, které mohou být nahrazeny v jednom výkyvu. Hodnota o je algoritmem zvyšována po neúspěšných oscilačních cyklech a resetována na hodnotu 1 po každém zlepšení X_d.



Obr. 13.10 Grafy demonstrují vývoj velikosti zkoumaných podmnožin d-parametrizovaných algoritmů výběru příznaků: a) sekvenční dopředný výběr, b) sekvenční dopředný plovoucí výběr, c) oscilační vyhledávání.

Algoritmus je ukončen když o překročí uživatelem stanovený *limit* Δ. Průběh oscilačního vyhledávání je ilustrován ve srovnání s SFS a SFFS na obr. 13.10.

Každý algoritmus OS vyžaduje nějakou počáteční množinu d příznaků. Tuto počáteční množinu lze získat náhodně či jakýmkoliv jiným způsobem, tj. užitím některé z tradičních sekvenčních vyhledávacích procedur. Dále pro nahrazení o-tic příznaků v horních a dolních výkyvech lze použít téměř libovolnou proceduru výběru příznaků.

Oscilační vyhledávání (OS)
Algoritmus generuje podmnožinu d příznaků s rozsahem prohledávání omezeným parametrem $\Delta \geq 1$ (pro hledání v plném rozsahu položme $\Delta = D$).

KROK 1: *Inicializace.* Vycházíme z existující podmnožiny d příznaků X_d, kterou můžeme získat libovolným způsobem. Nastavme *hloubku oscilačního cyklu* $o = 1$.

KROK 2: Vygenerujme podmnožinu $X_d^{\downarrow} = ADD^o(REM^o(X_d))$.

KROK 3: Je-li $X_d^{\downarrow}$ lepší než X_d, položme $X_d = X_d^{\downarrow}$, $o = 1$, a jděme na Krok 2.

KROK 4: Vygenerujme podmnožinu $X_d^{\uparrow} = REM^o(ADD^o(X_d))$.

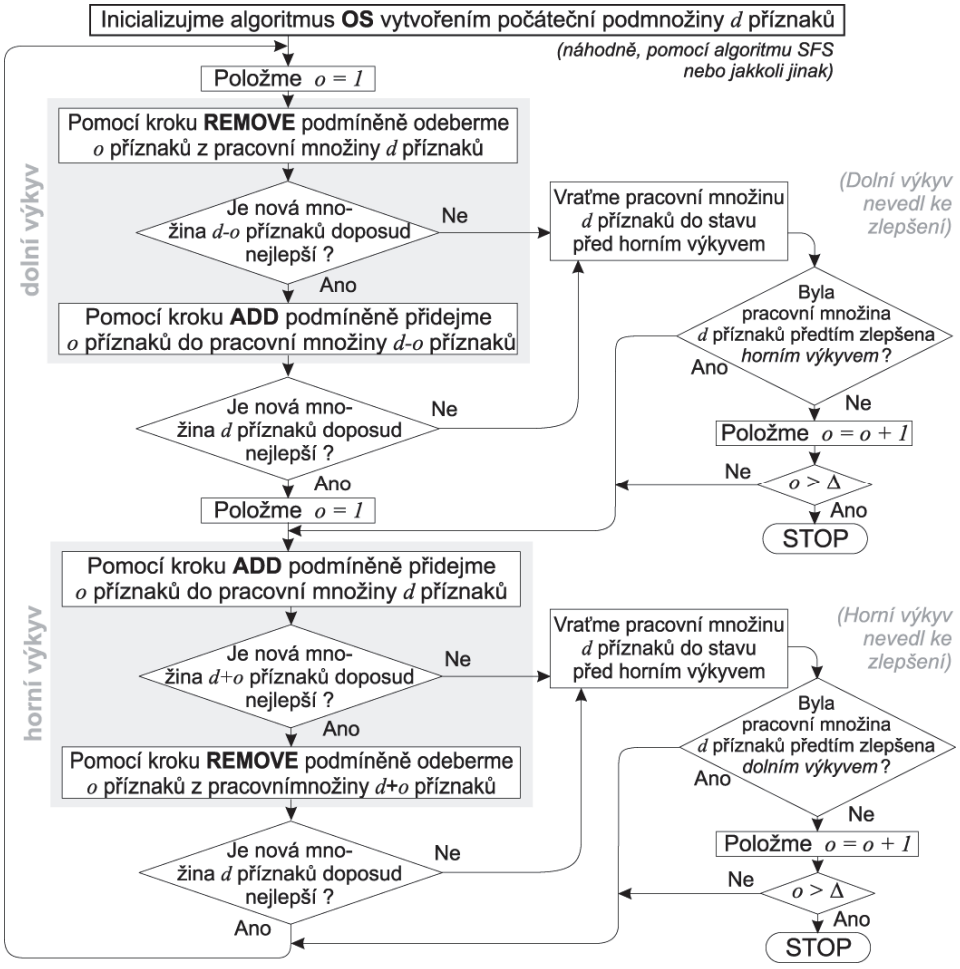
KROK 5: Pokud je $X_d^{\uparrow}$ lepší než X_d, položme $X_d = X_d^{\uparrow}$, $o = 1$, a jděme na Krok 2.

KROK 6: Pokud $o < \Delta$, položme $o = o + 1$ a jděme na Krok 2.

Obecnost konceptu vyhledávání OS umožňuje přizpůsobit algoritmus pro větší rychlost nebo naopak větší přesnost nastavením Δ, volbou inicializační procedury, popř. předefinováním operací *ADD* a *REM*. Na rozdíl od ostatních sekvenčních vyhledávacích procedur OS neztrácí čas vyhodnocováním podmnožin, jejichž kardinalita je příliš vzdá-

lena kardinalitě cílové (d). To zlepšuje schopnost OS nalézt dobrá řešení pro podmnožiny dané kardinality. Nejrychlejší zlepšování cílové podmnožiny lze očekávat v počátečních fázích algoritmu z důvodu malé počáteční hloubky oscilačních cyklů a tím i jejich rychlejšího průběhu. Později, když se stávající podmnožina příznaků přibližuje těsněji k optimu, cykly s malou hloubkou selhávají ve snaze zlepšit řešení a algoritmus proto rozšiřuje rozsah prohledávání ($o = o + 1$). I když se takto zvyšuje šance dostat se blíže k optimu, kompromis mezi šancí na nalezení lepšího řešení a růstem výpočetního času nabývá na významu. Toto chování algoritmu je nicméně velmi výhodné, protože dává možnost zastavit vyhledávání po určité omezené době, aniž by se tím riskovala ztráta výrazně lepšího výsledku. Hlavní výhody oscilačního vyhledávání lze shrnout takto:

– Lze na něj pohlížet jako na univerzální ladicí mechanismus schopný zlepšit řešení získaná jiným způsobem.



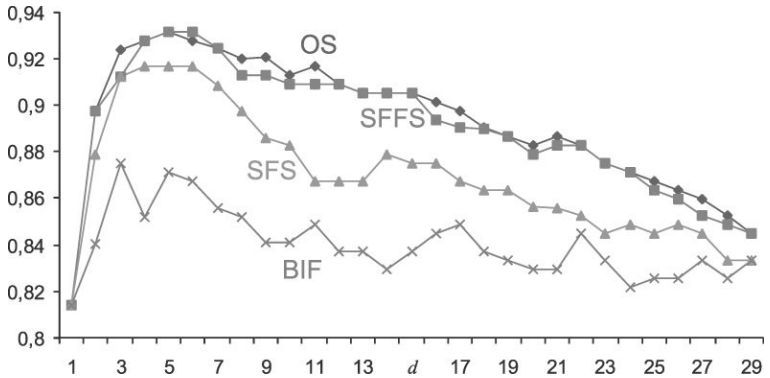
Obr. 13.11. Algoritmus oscilačního vyhledávání.

– Pro nízké hodnoty Δ a při použití rychlé inicializace (např. náhodné) je algoritmus OS velmi rychlý. V případě úloh o velké dimenzionalitě je jednou z mála alternativ k BIF. V práci (Novovičová a kol., 2006) je OS úspěšně použito při kategorizaci dokumentů k výběru řádově stovek příznaků při dimenzionalitě problému 10 000.
– Protože OS zpracovává od samého počátku podmnožiny o cílové kardinalitě, může nalézt řešení dokonce i v případech, v nichž *top-down* či *bottom-up* sekvenční procedury selžou kvůli numerickým problémům.
– Protože se řešení postupně zlepšuje po každém oscilačním cyklu s nejviditelnějším zlepšením v počátku, je možné algoritmus předčasně ukončit po předem specifikované době výpočtu a přitom dostat použitelné řešení. Z tohoto důvodu je OS použitelné v real-time systémech.
– Standardní sekvenční metody jsou náchylné k uvíznutí v lokálních extrémech. Opakovaný běh OS z několika různých náhodných počátečních bodů vyhledávacího prostoru dává větší šanci se takovému lokálnímu extrému vyhnout.

13.5.7 Experimentální porovnání *d*-parametrizovaných metod

Suboptimální *d*-parametrizované metody výběru příznaků, jež jsme popisovali postupně v odstavcích 13.5.1, 13.5.2, 13.5.4 a 13.5.6, byly uvedeny postupně od jednoduchých po komplexní. Metoda BIF (odst. 13.5.1) je nejrychlejší ale zároveň nejslabší metodou z hlediska maximalizace kriteriální funkce. OS poskytuje z uvedených metod nejsilnější optimalizační schopnost za cenu nejpomalejšího výpočtu (v závislosti na nastavení). Pro ilustraci uvedeného chování porovnáme výstup BIF SFS SFFS a OS na vzorové úloze výběru příznaků. Uvedené metody použijeme jakožto pouzdra (viz odst. 13.3.2).

Každou z uvedených metod testujeme na datech *ionosphere* (34 dim., 2 třídy: 225 a 126 vzorků) získaných z databáze UCI repository (Asuncion a kol., 2007), vybíráme nejlepší podmnožiny příznaků všech velikostí d = 1,..., 34. Data byla rozdělena na dvě části: 80 % jako trénovací a 20 % testovací. Metody byly použity k maximalizaci přesnosti klasifikace 3-NN klasifikátoru (*k*-nejbližších sousedů, viz (Dasarathy, 1991)), přesnost byla odhadována pomocí 10násobné krosvalidace v rámci trénovací části dat vždy na zkoumaném podprostoru.

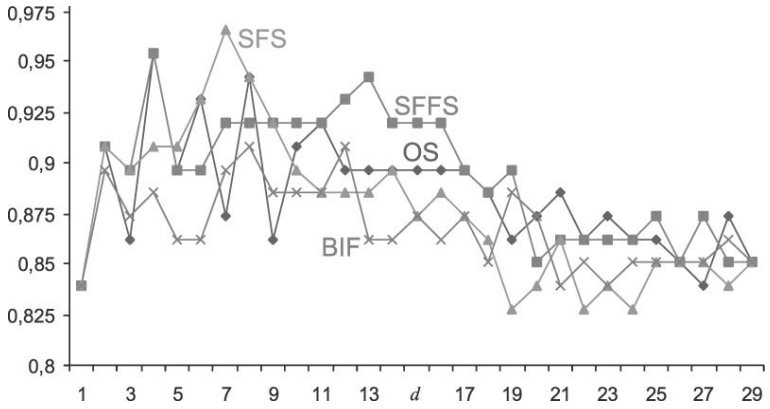


Obr. 13.12 Porovnání optimalizačního výkonu suboptimálních metod výběru příznaků (maximální dosažená úspěšnost klasifikace 3-NN na trénovacích datech).

BIF SFS a SFFS nevyžadují žádné parametry, OS bylo nastaveno tak, aby každé vyhledávání proběhlo 15krát z různých náhodných počátečních podmnožin dané velikosti s $\Delta = 15$. Toto nastavení je sice velmi náročné na výpočetní čas ale dává šanci vyhnout se lokálním extrémům, ve kterých skončí výpočet většiny vyhledávacích algoritmů.

Obrázek 13.12 ukazuje maximální hodnotu kritéria dosaženou jednotlivými metodami pro každou velikost podmnožiny. Je vidět, že nejlepší výsledky dává ve většině případů OS, i když SFFS zůstává pozadu jen nepatrně. Optimalizační schopnost SFS se ukazuje jako podstatně menší, ale stále ještě podstatně větší než u BIF.

Pro praktické použití je však důležitější schopnost generalizace. Obrázek 13.13 ukazuje vliv nalezených řešení na úspěšnost klasifikace nezávislých testovacích dat. Z tohoto hlediska se rozdíly mezi uvažovanými metodami výrazně stírají. Komplexnější metody se ukazují jako náchylnější k přetrénování (viz oddíl 13.9), v případě OS je zřejmý největší rozdíl mezi úspěšností klasifikace na trénovacích a testovacích datech. Metoda SFFS se zde ukazuje jako v průměru nejspolehlivější, SFS poskytuje v tomto příkladu nejlepší izolovaný nezávislý výsledek. Za povšimnutí stojí, že ačkoliv největší optimalizované kriteriální hodnoty byly dosaženy pro podmnožiny o velikosti zhruba 6 příznaků, nejlepší výsledky na nezávislých datech lze pozorovat pro podmnožiny o různých velikostech mezi 4 až 13 příznaky. Tento příklad dobře ilustruje jeden z klíčových problémů výběru příznaků, že je obtížné najít podmnožiny, které dobře zobecňují. Poznamenejme, že uvedené výsledky jsou ilustrační. Při užití odlišné kriteriální funkce lze očekávat také odlišnosti v detailech funkčnosti jednotlivých metod.

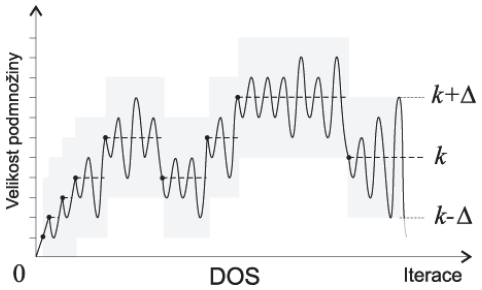


Obr. 13.13 Porovnání generalizační schopnosti suboptimálních metod (úspěšnosti klasifikace 3-NN na nezávislých datech).

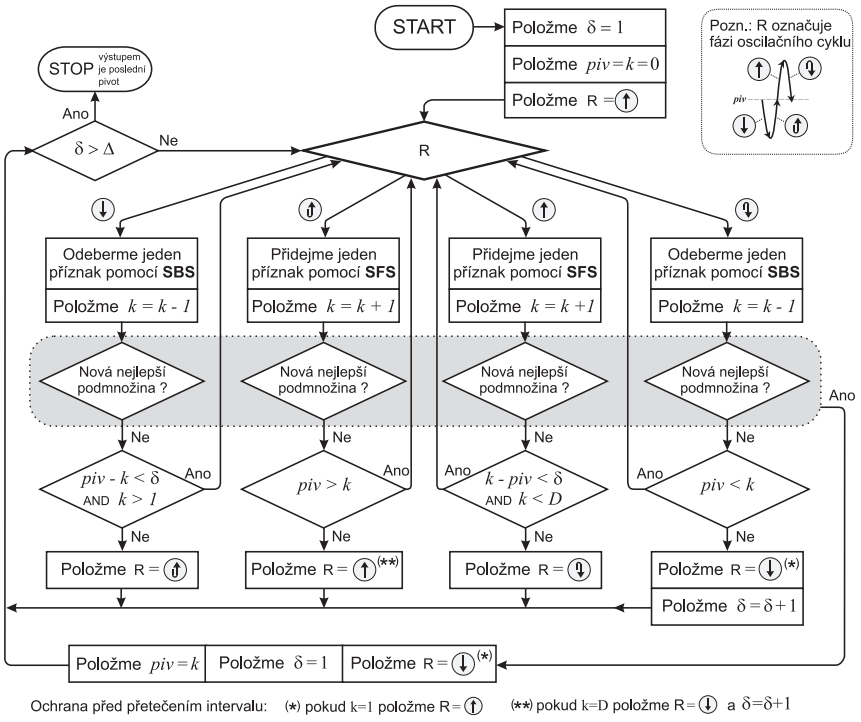
Rychlost každé z testovaných metod klesá s její složitostí. BIF má lineární výpočetní náročnost (ve vztahu k počtu vyhodnocených kritérií). Ostatní uvažované metody mají výpočetní náročnost polynomickou. SFFS je zde zhruba 10krát pomalejší než SFS. OS je při pomalém nastavení zhruba 10- až 100krát pomalejší než SFFS. Tyto údaje však silně závisí na konkrétních okolnostech řešené úlohy a zde slouží jen k nejhrubší orientaci.

13.6 Optimalizace velikosti podmnožiny – dynamické oscilační vyhledávání

Základní princip oscilačního vyhledávání (viz odst. 13.5.6) byl dále rozšířen do podoby *dynamického oscilačního vyhledávání* (Dynamic Oscillating Search, DOS) (Somol a kol., 2008b). Algoritmus DOS je iniciován libovolnou podmnožinou příznaků (včetně prázdné množiny). Podobně jako OS se opakovaně snaží zlepšit stávající množinu prováděním oscilačních cyklů (s rostoucí hloubkou oscilačního cyklu v případě neúspěchu).



Obr. 13.14 Graf demonstruje vývoj velikosti zkoumaných podmnožin *d*-optimalizujícího algoritmu výběru příznaků – dynamického oscilačního vyhledávání.



Obr. 13.15 Algoritmus dynamického oscilačního vyhledávání.

Velikost aktuálně vybrané podmnožiny se však v průběhu výpočtu mění a výpočet restartuje, kdykoliv je v libovolné fázi oscilačního cyklu nalezeno nové globálně nejlepší řešení. Na rozdíl od ostatních metod uváděných v této kapitole je tudíž DOS *d*-optimalizující procedurou.

Průběh dynamického oscilačního vyhledávání je ilustrován na obr. 13.14 (srovnejte s obr. 13.10). Podobně jako OS i DOS ukončuje výpočet, když stávající hloubka cyklu překročí uživatelem definovaný *limit* Δ. DOS také sdílí s OS stejné přednosti shrnuté v odst. 13.5.6, tedy schopnost doladit výsledky získané jiným způsobem, sekvenčnost zlepšování výsledku, nejrychlejší zlepšování v počátečních fázích vyhledávání atd. Blokové schéma algoritmu DOS vidíme na obr. 13.15.

Dynamické oscilační vyhledávání (DOS)
Algoritmus generuje podmnožinu příznaků optimalizované velikosti k s rozsahem prohledávání omezeným parametrem $\Delta \geq 1$ (pro hledání v plném rozsahu položme $\Delta = D$).

KROK 1: *Inicializaci* proveďme jedním ze dvou způsobů. Nastavme *hloubku oscilačního cyklu* $\delta = 1$. Dále

a) spočtěme výchozí podmnožinu $X_k = ADD(ADD\ (\emptyset))$, $k = 2$, nebo

b) můžeme vyjít z libovolné existující podmnožiny k příznaků X_k, pokud $k \geq 2$.

KROK 2: Proveďme výpočet $ADD^{\delta}(REM^{\delta}(X_k))$, pokud je kterákoliv přechodná podmnožina $X^{\downarrow}\ |X^{\downarrow}|= i$, $i \in [k - \delta, k]$, generovaná v průběhu výpočtu shledána lepší než X_k, položme $X_k = X^{\downarrow}$, $k = i$, $\delta = 1$, a restartujme Krok 2.

KROK 3: Proveďme výpočet $REM^{\delta}(ADD^{\delta}(X_k))$, pokud je kterákoliv přechodná podmnožina $X^{\uparrow}\ |X^{\uparrow}|= j$, $j \in [k, k + \delta]$, generovaná v průběhu výpočtu shledána lepší než X_k, položme $X_k = X^{\uparrow}$, $k = j$, $\delta = 1$, a jděme na Krok 2.

KROK 4: Pokud $\delta < \Delta$, položme $\delta = \delta + 1$ a jděme na Krok 2.

V průběhu vyhledávání generuje DOS posloupnost řešení se vzrůstajícími hodnotami kritéria a za předpokladu, že hodnota kritéria neklesá se zmenšující se velikostí podmnožiny. Kompromis mezi časem vyhledávání a blízkostí k optimálnímu řešení lze řídit předčasným přerušením vyhledávání. Počet výpočtů kritéria je řádu $O(n^3)$. Celková doba vyhledávání však silně závisí na zvolené hodnotě Δ, na konkrétních datech a vlastnostech kritéria a konečně na nepredikovatelném počtu restartů oscilačních cyklů, ke kterým dochází po každém nalezení lepšího řešení.

13.6.1 Experimentální porovnání *d*-optimalizujících metod

Abychom mohli experimentálně porovnat algoritmus DOS s dříve diskutovanými metodami SFS SFFS a OS, použijeme tyto metody v *d*-optimalizující úpravě, takto bude každá z uvedených metod volána pro všechny možné velikosti podmnožin, ze získaných výsledků všech kardinalit je poté jako výstup dané metody vybrán výsledek s největší hodnotou kritéria. Abychom vyznačili tento odlišný způsob použití dříve diskutovaných metod, označíme příslušné metody SFS* SFFS* a OS*.

Porovnání provedeme na mamogramových datech *wdbc* (30 dim., 2 třídy: 357 benigních a 212 maligních vzorků) z databáze UCI Repository (Asuncion a kol., 2007) při použití tří různých kritérií. Budeme vyhodnocovat přesnost klasifikace bayesovského klasifikátoru pro normální rozložení klasifikátoru 3-NN (Dasarathy, 1991) a SVM (Sup-

port Vector Machine) s jádrem radiální bázové funkce, viz (Chang a kol., 2001). Experimenty byly provedeny s použitím dvoustupňové krosvalidace. Vnější krosvalidační cyklus poskytuje různá dělení dat na trénovací a testovací část. Vnitřní krosvalidační cyklus pak dále rozděluje trénovací část dat na část pro trénink a validaci klasifikátoru. Výsledky experimentů jsou shrnuty v tabulce 13.3. Tabulka obsahuje tři oddíly udávající výsledky pro jednotlivé klasifikátory (kriteriální funkce). Sloupec I-CV ukazuje pro každou metodu výběru příznaků maximum dosažené hodnoty kritéria (průměr přes vnitřní krosvalidační cyklus). Sloupec O-CV vyjadřuje klasifikační přesnost na nezávislých testovacích datech (průměr přes vnější krosvalidační cyklus).

Tabulka 13.3 Porovnání d-optimalizujících suboptimálních metod (optimalizační výkonnost I-CV, generalizační schopnost O-CV, velikost nalezených podmnožin)

kritérium	metoda výběru	max. hodn. krit. (I-CV)	úspěšnost klasif. (O-CV)	velikost podmn.	doba výpočtu
bayesovský klasifikátor	SFS*	0,962	0,933	10,8	00:00
	SFFS*	0,972	0,942	10,6	00:03
	OS*	0,970	0,940	9,9	00:06
	DOS	0,973	0,951	10,7	00:06
	všechny příznaky		0,945	30	
3-NN	SFS*	0,981	0,967	15,3	00:01
	SFFS*	0,983	0,970	13,7	00:09
	OS*	0,982	0,965	14,2	00:22
	DOS	0,984	0,965	12,4	00:31
	všechny příznaky		0,972	30	
SVM	SFS*	0,979	0,970	18,5	00:05
	SFFS*	0,982	0,968	16,2	00:23
	OS*	0,981	0,974	16,7	00:58
	DOS	0,983	0,968	12,8	01:38
	všechny příznaky		0,972	30	

Z výsledků lze vypozorovat, že dynamické oscilační vyhledávání je schopné překonat ostatní testované metody z hlediska schopnosti maximalizace kritéria (I-CV), má tendenci produkovat nejmenší podmnožiny příznaků, ale podobně jako v případě *d*-parametrizovaných experimentů (odst. 13.5.7) je zřejmé, že žádná z uvažovaných metod není jednoznačně nejlepší z hlediska generalizační schopnosti (úspěšnosti klasifikace na nezávislých datech O-CV).

13.7 Hybridní algoritmy

V předchozích oddílech jsme podali stručný přehled nejrůznějších nástrojů vhodných pro řešení úlohy výběru příznaků. Poukázali jsme také na fakt, že žádnou z uvedených metod nelze označit jako univerzálně nejlepší. Různé metody jsou více či méně vhodné v různých situacích. Příkladem může být rozdíl mezi filtry a pouzdry (viz odst. 13.3.2), filtry bývají podstatně rychlejší než pouzdra a umožňují tak řešit problémy podstatně větší dimenzionality, pouzdra však lépe umožňují při výběru příznaků vzít v úvahu

vlastnosti konkrétního klasifikátoru a mohou tudíž dosáhnout lepší klasifikační přesnosti. V takovéto situaci se nabízí idea zkombinovat vlastnosti obou uvažovaných přístupů.

Jednoduchý způsob definice *hybridní metody* výběru příznaků, kombinující výhody jiných existujících přístupů, byl popsán v práci (Liu a kol., 2005). Obecněji použitelný mechanismus definice hybridní metody, umožňující kombinovat libovolné dva způsoby vyhodnocování kriteriální, funkce byl ukázán v článku (Somol a kol., 2006). V hybridních sekvenčních metodách definovaných tímto způsobem přepokládáme, že v každém kroku algoritmu je o přidání vhodného příznaku do pracovní množiny či odebrání z ní ven rozhodováno dvoufázově. Nejprve je využito rychlé *předfiltrovací kriteriální funkce* k redukci počtu kandidátů teprve, poté je mezi menším počtem kandidátů výsledný příznak vybrán pomocí pomalejší, ale jak předpokládáme vhodnější *hlavní kriteriální funkce*. Toto schéma lze implementovat v libovolných sekvenčních algoritmech výběru příznaků nahrazením Definice 1 a Definice 2 (kroky *ADD* a *REM* uvedené v odst. 13.5.2) upravenými definicemi, které uvádíme v dalším textu. Pro zjednodušení označme $J_F(.)$ rychleji vyhodnotitelné, ale pro daný problém méně vhodné *předfiltrovací kritérium*, dále označme $J_W(.)$ pomaleji vyhodnotitelné, ale pro danou úlohu předpokládané vhodnější *hlavní kritérium*. Dále zavedeme *hybridizační koeficient* $\lambda \in [0, 1]$ jakožto uživatelský parametr určující relativně podíl kandidátských příznaků zachovávaných po filtraci předběžným kritériem a následně vyhodnocovaných pomocí primárního kritéria. Poznamenejme, že v následujících definicích $\lceil \cdot \rceil$ označuje zaokrouhlení hodnoty.

Definice 3. Pro stávající množinu příznaků X_d a dané $\lambda \in [0, 1]$ nechť Z^+ je množina kandidátských příznaků

$$Z^+ = \{f_i : f_i \in Y \setminus X_d ; i = 1,..., \max\{1, \lceil \lambda \cdot | Y \setminus X_d | \rceil\}\} \tag{13.14}$$

taková že

$$\forall f, g \in Y \setminus X_d, f \in Z^+, g \notin Z^+, \quad J_F^+(X_d, f) \geq J_F^+(X_d, g), \tag{13.15}$$

kde $J_F^+(X_d, f)$ označuje předfiltrovací kriterální funkci použitou pro vyhodnocení podmnožiny získané přidáním f do X_d, když $f \in Y \setminus X_d$. Nechť f^+ je takový příznak, že

$$f^+ = \arg\max_{f \in Z^+} J_W^+(X_d, f), \tag{13.16}$$

kde $J_W^+(X_d, f)$ označuje hlavní kriteriální funkci použitou pro vyhodnocení podmnožiny získané přidáním f do X_d, je-li $f \in Z^+$. Pak budeme říkat, že $ADD_H(X_d)$ je operace přidání příznaku f^+ do množiny X_d, abychom získali množinu X_{d+1}, jestliže

$$ADD_H(X_d) \equiv X_d \cup \{f^+\} = X_{d+1}, \quad X_d, X_{d+1} \subset Y. \tag{13.17}$$

Definice 4. Pro stávající množinu X_d a dané $\lambda \in [0, 1]$ nechť Z^- je množina kandidátských příznaků

$$Z^- = \{f_i : f_i \in X_d ; i = 1,..., \max\{1, \lceil \lambda \cdot | X_d | \rceil\}\} \tag{13.18}$$

taková, že

$$\forall f, g \in X_d, f \in Z^-, g \notin Z^-, \quad J_F^-(X_d, f) \geq J_F^-(X_d, g) \ , \tag{13.19}$$

kde $J_F^-(X_d, f)$ označuje předfiltrovací kriteriální funkci použitou pro vyhodnocení podmnožiny získané vyloučením f z X_d, je-li $f \in X_d$. Nechť f^- je takový příznak, že

$$f^- = \arg\max_{f \in Z^-} J_W^-(X_d, f), \tag{13.20}$$

kde $J_W^-(X_d, f)$ označuje hlavní kriteriální funkci použitou pro vyhodnocení podmnožiny získané vyloučením f z X_d, když $f \in Z^-$. Pak budeme říkat, že $REM_H(X_d)$ je operace odstranění příznaku f^- z množiny X_d, abychom získali množinu X_{d-1}, jestliže

$$REM_H(X_d) \equiv X_d \setminus \{f^-\} = X_{d-1}, \quad X_d, X_{d-1} \subset Y. \tag{13.21}$$

Efekt hybridizace je ilustrován na příkladu z tabulky 13.4. Testovali jsme hybridizovanou metodu DOS (viz oddíl 13.6) na datech *waveform* (40 dim., 2 třídy: 1692 a 1653 vzorků) z databáze UCI Repository (Asuncion a kol., 2007). Jakožto předfiltrovací kritérium bylo použito Bhattacharyovy vzdálenosti (Devijver a kol., 1982). Hlavním kritériem byl odhad úspěšnosti klasifikace 3-NN klasifikátoru (*k*-Nearest Neigbor, neboli *k*-nejbližších sousedů, viz (Dasarathy, 1991)).

Uvedené výsledky ukazují nárůst získané hodnoty kritéria, ale také výrazný nárůst doby výpočtu pro rostoucí hodnoty λ, jenž odpovídá rozšiřování počtu kandidátských příznaků vyhodnocovaných pomocí hlavního kritéria. Ověřením na nezávislých datech se však v tomto případě ukázalo, že hybridizací lze nejen ušetřit výpočetní čas, ale i zlepšit chování klasifikátoru na neznámých datech. Nejlepší výsledek zde odpovídá nejnižší hodnotě λ. Je zřejmé že v daném případě vede maximalizace hlavního kritéria metodou DOS k přeučení a že předfiltrovací kritérium zde přeučení úspěšně omezuje.

Tabulka 13.4 Porovnání úspěšnosti hybridizovaného algoritmu dynamického oscilačního vyhledávání pro různé koeficienty hybridizace

koeficient hybridizace λ	0,01	0,25	0,5	0,75	1
maximální hodnota kritéria	0,907	0,913	0,921	0,921	0,921
přesnost klasifikace na nezáv. datech	0,916	0,911	0,911	0,910	0,910
nalezená velikost podmnožiny	11	10	15	17	17
výpočetní čas	1:12	8:06	20:42	35:21	48:24

13.8 Výběr příznaků založený na modelu směsi hustot

Alternativou k vyhledávacím metodám výběru příznaků typu filtr či pouzdro (viz odst. 13.3.2) jsou tzv. metody integrované (embedded), v nichž je proces výběru příznaků přímo součástí učícího procesu, resp. odhadu modelu.

Uvažujme nyní problém výběru příznaků v situacích, kdy vícerozměrné podmíněné hustoty $p(\boldsymbol{x} \mid \omega_j)$ a často i apriorní pravděpodobnosti $P(\omega_j)$ jsou buď jen částečně známé, nebo neznámé a jediným zdrojem informace se stávají soubory nezávislých pozorování z jednotlivých tříd, tj. máme k dispozici trénovací množiny T_ω se vzorky ze třídy ω.

Odhadování apriorních pravděpodobností většinou není problém, obvykle je nahrazujeme relativními četnostmi výskytu vektorů příznaků z jednotlivých tříd v trénovací množině, nebo známe odhady jejich výskytu v praxi. Některé nebo všechny podmíněné hustoty je vhodné modelovat jako konečné směsi hustot pravděpodobnosti, které jsou flexibilnější než jednoduché pravděpodobnostní modely. Pomocí modelu směsí lze reprezentovat libovolně složité hustoty, viz např. (Hastie a kol., 1996), (Palm, 1994), (McLachlan, 2004), (Ripley, 2005) a (Webb, 2002).

V publikacích (Pudil a kol., 1994b), (Pudil a kol., 1995), (Novovičová a kol., 1996) a (Novovičová a kol., 1998) je uveden přístup k výběru příznaků založený na aproximaci neznámých podmíněných hustot pravděpodobnosti modifikovanou konečnou směsí součinových komponent.

13.8.1 Konečná směs hustot pravděpodobnosti

Konečná směs hustot pravděpodobnosti pro podmíněnou hustotu ve třídě ω je definována ve tvaru

$$p(\boldsymbol{x} \mid \alpha_\omega, \theta_\omega) = \sum_{m=1}^{M_\omega} \alpha_m^\omega \, p(\boldsymbol{x} \mid \theta_m^\omega), \quad \sum_{m=1}^{M_\omega} \alpha_m^\omega = 1, \quad \alpha_m^\omega > 0, \tag{13.22}$$

kde M_ω označuje počet komponent směsi a $p(\boldsymbol{x} \mid \theta_m^\omega)$ je hustota m-té komponenty ve třídě ω, α_m^ω udává váhu m-té komponenty směsi, θ_m^ω značí parametr m-té komponenty směsi a $\{\alpha_\omega, \theta_\omega\} = \{\alpha_1^\omega, ..., \alpha_{M_\omega}^\omega, \theta_1^\omega, ..., \theta_{M_\omega}^\omega\}$ vyjadřuje množinu všech parametrů směsi. Tento model předpokládá, že každá komponenta směsi má vícerozměrné rozdělení s vlastními parametry. Obvykle uvažujeme, že komponenty mají stejný parametrický tvar (např. vícerozměrný normální). Model (13.22) může být zjednodušen za předpokladu, že pro každou komponentu $p(\boldsymbol{x} \mid \theta_m^\omega)$ směsi (13.22) jsou příznaky statisticky nezávislé. Pak je každá komponenta součinem jednorozměrných marginálních hustot $p(x_i \mid \theta_m^\omega)$ a směs (13.22) lze přepsat do jednoduššího tvaru

$$p(\boldsymbol{x} \mid \alpha_\omega, \theta_\omega) = \sum_{m=1}^{M_\omega} \alpha_m^\omega \prod_{i=1}^{D} p(x_i \mid \theta_{mi}^\omega), \tag{13.23}$$

v němž θ_{mi}^ω je parametr m-té komponenty ve třídě ω odpovídající příznaku x_i. To znamená že $p(x_i \mid \theta_m^\omega)$ je hustota i-tého příznaku v m-té komponentě hustoty ve třídě ω a podmíněná hustota pravděpodobnosti je modelována jako směs nezávislých pravděpodobnostních modelů.

13.8.2 Modifikovaná konečná směs součinových komponent

V popisovaném přístupu ke globálnímu výběru příznaků předpokládáme, že některé příznaky nejsou významné pro strukturu směsi (13.23) v tom smyslu, že příznak x_i je *nevýznamný* pro strukturu směsi (13.23), jestliže jeho rozdělení pravděpodobnosti je určeno *společnou* (*sdílenou*) *hustotou* napříč všemi komponentami, a tudíž je nezávislé na m, tj. $p(x_i \mid \theta_m^\omega) = g_0(x_i \mid \theta_{0i})$. Příznak x_i je *významný* pro strukturu směsi (13.23),

jestliže jeho rozdělení pravděpodobnosti je určeno *specifickou hustotou*, tj. $p(x_i \mid \theta_m^\omega) = g_0(x_i \mid \theta_{mi}^\omega)$. Nechť $\Phi = (\phi_1, \phi_2, ..., \phi_D) \in \{0, 1\}^D$ je množina binárních parametrů a dále

$\phi_i = 1$ znamená, že příznak x_i je *významný* a měl by být modelován použitím *specifické komponenty* směsi,

$\phi_i = 0$ znamená, že příznak x_i je *nevýznamný* a měl by být reprezentován *společnou (sdílenou) hustotou*.

Směs hustot v (13.23) lze nyní přepsat do tvaru

$$p(\boldsymbol{x} \mid \alpha_\omega, \theta_\omega, \theta_0, \Phi) = \sum_{m=1}^{M_\omega} \alpha_m^\omega p(\boldsymbol{x} \mid \theta_\omega, \theta_0, \Phi) = \sum_{m=1}^{M_\omega} \alpha_m^\omega \prod_{i=1}^{D} [g_0(x_i \mid \theta_{0i})^{1-\phi_i} g(x_i \mid \theta_{mi}^\omega)^{\phi_i}], \quad (13.24)$$

kde $\theta_0 = (\theta_{01}, ..., \theta_{0D})$. Takto lze pohlížet na binární parametry ϕ_i jako na *řídicí veličiny*, neboť složitost a struktura směsi (13.23) může být řízena pomocí těchto parametrů.

Výpočet odhadů parametrů $(\alpha, \theta, \theta_0, \Phi) = \{(\alpha_\omega, \theta_\omega, \theta_0, \Phi, \omega \in \Omega\}$ směsi (13.24) maximalizujících pro dané trénovací množiny T_ω, $\omega \in \Omega$, globální logaritmickou věrohodnostní funkci

$$L(\alpha, \theta, \theta_0, \Phi) = \sum_{\omega \in \Omega} \frac{P(\omega)}{|\mathcal{T}_\omega|} \sum_{\mathbf{x} \in \mathcal{T}_\omega} \sum_{m=1}^{M_\omega} \log p(\boldsymbol{x} \mid \alpha_\omega, \theta_\omega, \theta_0, \Phi) \quad (13.25)$$

je založen na algoritmu EM (Expectation-Maximization) (Redner a kol., 1984).

• *E-krok*: Pro dané parametry $\alpha_m^\omega, \theta_m^\omega, \theta_0, \Phi$ a pro $m = 1, ..., M_\omega$, $\boldsymbol{x} \in T_\omega$, vypočteme aposteriorní pravděpodobnosti

$$p(m \mid \boldsymbol{x}, \omega) = \frac{\alpha_m^\omega p(\boldsymbol{x} \mid \theta_m^\omega, \theta_0, \Phi)}{\sum_{j=1}^{M_\omega} \alpha_j^\omega p(\boldsymbol{x} \mid \theta_j^\omega, \theta_0, \Phi)} = \frac{\alpha_m^\omega \prod_{i=1}^{D} [g_0(x_i \mid \theta_{0i})^{1-\phi_i} g(x_i \mid \theta_{mi}^\omega)^{\phi_i}]}{\sum_{j=1}^{M_\omega} \alpha_j^\omega \prod_{i=1}^{D} [g_0(x_i \mid \theta_{0i})^{1-\phi_i} g(x_i \mid \theta_{ji}^\omega)^{\phi_i}]}. \quad (13.26)$$

Pro tyto pravděpodobnosti určíme očekávanou globální logaritmickou věrohodnostní funkci

$$L(\alpha, \theta, \theta_0, \Phi) = \sum_{\omega \in \Omega} \frac{P(\omega)}{|\mathcal{T}_\omega|} \sum_{\mathbf{x} \in \mathcal{T}_\omega} \sum_{m=1}^{M_\omega} p(m \mid \boldsymbol{x}, \omega) \log \alpha_m^\omega p(\boldsymbol{x} \mid \theta_\omega, \theta_0, \Phi). \quad (13.27)$$

• *M-krok*: Při pevných aposteriorních pravděpodobnostech (13.26) vypočteme nové hodnoty parametrů směsi

$$(\hat{\alpha}, \hat{\theta}, \hat{\theta}_0, \hat{\Phi}) = \arg \max_{\alpha, \theta, \theta_0, \Phi} \{L(\alpha, \theta, \theta_0, \Phi)\}. \quad (13.28)$$

Odhad binárních parametrů ϕ_i, $i = 1, ..., D$, je zahrnut do M-kroku algoritmu EM a závisí na zvoleném kritériu významnosti příznaků.

Poznamenejme, že v algoritmu EM předpokládáme na počátku iniciované hodnoty parametrů. Obvyklým způsobem inicializace je nějaká forma náhodné inicializace, podrobnější údaje najdeme v práci (McKenzie a kol., 1994).

Výpočet se značně zjednoduší za předpokladu že:
– jednorozměrné hustoty $g(x_i \mid \theta_{mi}^{\omega})$ a $g_{0i}(x_i \mid \theta_{0i})$ patří do rodiny jednorozměrných normálních hustot,
– $g_{0i}(x_i \mid \theta_{0i})$ má stejné parametry pro všechny třídy $\omega \in \Omega$,
– počet komponent směsi (13.24) a počet příznaků je předem stanoven.

Protože zde není možné prezentovat úplný podrobný popis zmíněných metod, odkazujeme čtenáře na původní zdroje (Pudil a kol., 1995), (Novovičová a kol., 1996). Základní myšlenka modelu (13.24) inspirovala řadu odborníků k dalšímu vývoji tohoto postupu, např. (Miller a kol., 2003), (Law a kol., 2004), (Krishnapuram a kol., 2004), (Graham a kol., 2006), (Bouguila a kol., 2009) a (Bouguila, 2010).

13.8.3 Míry významnosti příznaků při užití směsového modelu

Navržený model směsi (13.24) umožňuje odvodit dvě metody výběru příznaků, které se liší volbou kritéria vyhodnocování významnosti příznaků.

Aproximační metoda pomocí odhadnutých řídicích veličin, tj. vektoru $\hat{\Phi}_d$ s d nenulovými složkami, vyhledá příznaky, které dávají množinu aproximací podmíněných hustot pravděpodobnosti nejlepší ve smyslu minimalizace směsi Kullbackových–Leiblerových vzdáleností mezi skutečnou a předpokládanou podmíněnou hustotou vektoru $\boldsymbol{x}$ ve třídě ω s váhami $P(\omega_1),\ldots, P(\omega_C)$. Příznaky jsou vybírány s ohledem na reprezentační schopnost, která v některých případech nemusí znamenat diskriminační schopnost.

Divergenční metoda určí ty příznaky, které jsou nejlepší ve smyslu maximalizace Kullbackovy J-divergence (Boekee a kol., 1979) mezi dvěma třídami. Cílem metody je vybrat ty příznaky, které jsou nejlepší ve smyslu rozlišení mezi třídami.

13.8.4 Odvození rozhodovacího pravidla a souhrn vlastností směsového přístupu

Popsali jsme řešení problému výběru příznaků, při kterém místo výběru vhodné podmnožiny příznaků pomocí vyhledávacích metod odhadujeme množinu binárních veličin (jednu pro každý příznak), jež charakterizují významnost každého příznaku. Tento odhad je proveden pomocí algoritmu EM.

Uvažujme problém klasifikace daného objektu popsaného vektorem $\boldsymbol{x}$ do jedné třídy z množiny Ω. To je možné pomocí Bayesova rozhodovacího pravidla s minimální chybou:

Zařaď $\boldsymbol{x}$ do třídy ω_l, jestliže

$$\omega_l = \arg\max_{j=1,\ldots,C} \frac{P(\omega_j)\,p(\boldsymbol{x} \mid \omega_j)}{p(\boldsymbol{x})},\ l \in \{1,\ldots,C\},\quad p(\boldsymbol{x}) = \sum_{j=1}^{C} P(\omega_j)\,p(\boldsymbol{x} \mid \omega_j), \tag{13.29}$$

kde $p(\boldsymbol{x})$ je nepodmíněná hustota pravděpodobnosti rozložení vektoru $\boldsymbol{x}$. Dosadíme-li v uvedeném smyslu nalezenou optimální aproximaci podmíněné hustoty do Bayesova pravidla (13.29), společná hustota g_0 může být eliminována a nový objekt reprezentovaný vektorem $\boldsymbol{x}$ lze klasifikovat bezprostředně do jedné z možných tříd pouze na základě d příznaků $x_{i_1},\ldots,x_{i_d}$, kde $\{i_1,\ldots, i_d\}$ je permutace čísel $\{1,\ldots, D\}$, a to určením odhadu vektoru $\hat{\Phi}_d$ o d nenulových složkách. Tento klasifikátor v redukovaném příznakovém prostoru nazýváme *pseudoBayesův* klasifikátor.

Popsaný přístup k výběru příznaků je vhodný zvláště pro případy, když máme k dispozici dostatečně velkou trénovací množinu a zároveň některé nebo všechny podmíněné hustoty pravděpodobnosti jsou vícemodální, nebo když se předpokládá existence podtříd v každé třídě. Popsaný přístup může sloužit k více cílům:
– určení struktury vícerozměrných rozdělení,
– určení nejdůležitějších příznaků a tím možnosti snížit dimenzionalitu,
– odvození rozhodovacího pravidla založeného na vybraných příznacích pro řešení problému klasifikace do dvou nebo více tříd.

13.8.5 Experiment na reálných datech

Za pomoci metod výběru příznaků založených na modelu směsi (13.24) jsme provedli experimenty na datech *speech* a *wdbc* použitých již v odstavcích 13.4.6.2 a 13.6.1. V tomto experimentu bylo 67 % dat použito pro trénink a 33 % pro test úspěšnosti klasifikace. Experimenty jsme uskutečnili při dvou různých způsobech inicializace: náhodné inicializaci a inicializaci typu psi a králíci (McKenzie a kol., 1994). Tabulka 13.5 demonstruje potenciál směsové aproximační metody – s 5 komponentami (sloupec aprox. 5 k.) pro data *speech* a s 1, 5 či 20 komponentami pro data *wdbc*, metoda dosahuje lepší úspěšnosti klasifikace než gaussovský klasifikátor, který je založen na předpokladu unimodality dat.

Tabulka 13.5 Porovnání úspěšnosti klasifikace gaussovského a pseudobayesovského klasifikátoru (pro aproximační metodu) za použití různých počtů komponent směsi

		gauss. klasif.	aprox. 1 k.	aprox. 5 k.	aprox. 10 k.	aprox. 20 k.
speech	(náhodná inicializace)	0,916	0,784	0,924	0,908	0,910
	(inicializace psi a králíci)	–	0,784	0,926	0,936	0,916
wdbc	(náhodná inicializace)	0,940	0,947	0,947	0,940	0,954
	(inicializace psi a králíci)	–	0,947	0,947	0,940	0,940

13.9 Problém přeučení a problém stability výběru příznaků

Ve starší literatuře bylo zvykem ohodnocovat účinnost metod výběru příznaků na základě schopnosti najít optimum nebo hodnotu co nejbližší k optimu vzhledem ke zvolené kriterální funkci. V poslední době je důraz kladen na vyhodnocení schopnosti generalizovat, tj. schopnosti nalézt takové příznaky, pro něž výsledné rozhodovací pravidlo dává nejlepší výsledky na nezávislých datech. Bylo ukázáno, že podobně jako při konstrukci klasifikátorů je v případě výběru příznaků nutné zabránit přetrénování, tj. přílišnému přizpůsobení výsledku detailním vlastnostem trénovacích dat, které však mohou být zavádějící, zejména nemají-li trénovací data dostatečnou velikost (Raudys, 2006). Problém přetrénování je ilustrován poklesem úspěšnosti klasifikace na obr. 13.12 v porovnání s obrázkem 13.13. Testování účinnosti výběru příznaků na nezávislých testovacích datech je důležité také při porovnávání jednotlivých metod (Reunanen, 2003).

Problém porovnávání metod výběru příznaků mezi sebou je v literatuře chápán dvojznačně. Je rozdíl, porovnáváme-li samotnou optimalizační schopnost vyhledávacích procedur nebo účinnost klasifikátoru při použití metod výběru ve specifickém kontextu.

Konečná účinnost klasifikátoru je nejdůležitější mírou kvality, je však kontextově závislá a obvykle není podle ní vhodné přijímat zobecňující závěry o jednoznačné nadřazenosti či podřazenosti jedněch metod oproti druhým.

V literatuře panuje obecná shoda, že výběr příznaků založený na pouzdrech vede k přesnějším klasifikátorům než výběr příznaků založený na filtrech (viz odst. 13.3.2). Zároveň je však třeba zdůraznit, že pouzdra bývají náchylnější k přetrénování. Naopak slabší vztah mezi kriteriálními funkcemi založenými na filtrech a přesnosti konkrétního klasifikátoru může pomoci k lepšímu zobecnění a tím k lepší funkčnosti klasifikátoru na neznámých datech. Podobně lze očekávat větší robustnost výsledků filtrového výběru příznaků při použití v kontextu více různých klasifikátorů.

Problém odhadu účinnosti klasifikátoru není v žádném případě jednoduchý. Existuje mnoho strategií odhadování, jejichž vhodnost je závislá na problému (resubstituce, dělení dat, metody hold-out, krosvalidace, metoda leave-one-out atd.). Podrobnou studií o problémech souvisejících s učením klasifikátorů stabilizací slabých klasifikátorů a vyhodnocováním jejich přesnosti je (Skurichina, 2001).

13.9.1 Problém stability výběru příznaků

Výběr příznaků může být zatížen také problémem stability. Zdánlivě uspokojivý výsledek výběru příznaků tak může být ve skutečnosti výsledkem náhody, nestabilní metoda pak může při nepatrné změně okolností vést k výsledku zcela odlišnému. Je zřejmé, že nestabilní účinnost procesu FS by mohla vážně zhoršit vlastnosti konečného klasifikátoru výběrem špatných příznaků.

V souladu s prací (Kalousis a kol., 2007) definujeme *stabilitu algoritmu* výběru příznaků jakožto robustnost preferencí příznaků proti vlivu různých trénovacích množin generovaných ze stejného rozdělení. Stabilitu algoritmů FS budeme zkoumat v podobě jejich preference příznaků přes vybrané podmnožiny $S \subseteq Y$.

Současné práce zabývající se stabilitou metod FS se hlavně zaměřují na různé indexy stability zavádějící míry založené na Hammingově vzdálenosti (Dunne a kol., 2002), korelačních koeficientech a Tanimotově vzdálenosti (Kalousis a kol., 2007), indexu konzistence (Kuncheva, 2007) a Shannonově entropii (Křížek a kol., 2007). Poznamenejme, že stabilita procedury FS závisí na velikosti trénovací množiny, kritériu použitém k ohodnocení výběru příznaků a složitosti procedury FS (Raudys, 2006). V dalším výkladu se zaměříme na některé nové míry umožňující posoudit nejen stabilitu *d*-parametrizovaných ale i *d*-optimalizujících metod FS (Somol a kol., 2008a).

13.9.2 Vybrané míry stability výběru příznaků

Nechť $\mathsf{S} = \{S_1,..., S_n\}$je systém n podmnožin příznaků

$$S_j = \{f_{k_i} \mid i = 1,...,d_j, f_{k_i} \in Y, d_j \in \{1,...,|Y|\}\}, j = 1,...,n\ ,\ n > 1, n \in N\ ,$$

získaných z n opakovaných volání zkoumaného algoritmu výběru příznaků na různých trénovacích množinách vybraných z daného souboru dat. Naším cílem je definovat míru stability takovou, aby ohodnocovala systém S hodnotou z intervalu [0, 1], kde větší hodnota označuje větší stabilitu. Hodnota 0 nechť označuje situaci, kdy všechny podmnožiny příznaků generované zkoumanou metodou jsou disjunktní, a hodnota 1 nechť označuje situaci, kdy všechny podmnožiny příznaků jsou identické.

Nechť X je podmnožina množiny Y reprezentující všechny příznaky, které se objeví v kterékoliv množině v systému $\mathbf{S}$:

$$X = \{f \mid f \in Y, F_f > 0\} = \bigcup_{i=1}^{n} S_i, \;\; X \neq \emptyset, \tag{13.30}$$

kde F_f je počet výskytů (frekvence) příznaku $f \in Y$ v systému $\mathbf{S}$. Nechť N značí celkový počet výskytů každého příznaku v systému $\mathbf{S}$, tj.

$$N = \sum_{g \in X} F_g = \sum_{i=1}^{n} |S_i|, \;\; N \in \mathbf{N}, \;\; N \geq n. \tag{13.31}$$

Definice 5. Vážená konzistence $CW(\mathbf{S})$ systému $\mathbf{S}$ je definována jako

$$CW(\mathbf{S}) = \sum_{f \in X} w_f \frac{F_f - F_{\min}}{F_{\max} - F_{\min}}, \tag{13.32}$$

kde $w_f = \dfrac{F_f}{N}, \;\; 0 < w_f \leq 1, \sum_{f \in X} w_f = 1.$

Vzhledem k tomu, že $F_f = 0$ pro všechna $f \in Y \setminus X$, *vážená konzistence* $CW(\mathbf{S})$ může být ekvivalentně vyjádřena s použitím označení (13.31) ve tvaru

$$CW(\mathbf{S}) = \sum_{f \in X} \frac{F_f}{N} \cdot \frac{F_f - F_{\min}}{F_{\max} - F_{\min}} = \sum_{f \in Y} \frac{F_f}{N} \cdot \frac{F_f - 1}{n - 1}. \tag{13.33}$$

Je zřejmé, že $CW(\mathbf{S}) = 0$ tehdy a jen tehdy, když $N = |X|$, tj. tehdy a jen tehdy, když $F_f = 1$ pro všechna $f \in X$. Taková situace není ale myslitelná ve většině reálných úloh. Kdykoliv $n > |X|$, některé příznaky se musí objevit ve více než jedné podmnožině, a tudíž $CW(\mathbf{S}) > 0$. Podobně $CW(\mathbf{S}) = 1$ tehdy a jen tehdy, když $N = n|X|$, jinak všechny podmnožiny nemohou být identické.

Je zřejmé, že pro každé N, n reprezentující nějaký systém podmnožin $\mathbf{S}$ a pro dané Y existuje systém $\mathbf{S}_{\min}$ s takovou konfigurací příznaků ve svých podmnožinách, že dává minimální možnou hodnotu, označenou $CW_{\min}(N, n, Y)$, která může být větší než 0. Podobně existuje systém $\mathbf{S}_{\max}$, který dává maximální možnou hodnotu $CW(.)$, označíme ji $CW_{\max}(N, n)$, jež může být menší než 1.

Je zřejmé, že $CW_{\min}(.)$ je větší, pokud se velikosti podmnožin příznaků v systému blíží celkovému počtu příznaků $|Y|$, neboť v takovém systému jsou si podmnožiny navzájem podobnější. Proto použití míry (13.32) pro porovnání stability různých metod FS může vést k zavádějícím závěrům, mají-li porovnávané metody tendenci generovat systémy podmnožin výrazně odlišné průměrné velikosti. Tento problém budeme označovat jako problém zkreslení velikosti podmnožiny. Poznamenejme, že většina měr stability, které jsou k dispozici, je tímto problémem ovlivněna. Z tohoto důvodu zavádíme další míru nazvanou *relativní vážená konzistence*, která potlačuje vliv velikostí podmnožin v systému na výslednou hodnotu míry stability.

Definice 6. Relativní vážená konzistence $CW_{\mathrm{rel}}(\mathsf{S}, Y)$ systému S charakterizovaného veličinami N, n a daným Y je definovaná jako

$$CW_{\mathrm{rel}}(\mathsf{S},Y) = \frac{CW(\mathsf{S}) - CW_{\min}(N,n,Y)}{CW_{\max}(N,n) - CW_{\min}(N,n,Y)}, \tag{13.34}$$

kde $CW_{\mathrm{rel}}(\mathsf{S}, Y) = CW(\mathsf{S})$, pokud $CW_{\max}(N, n) = CW_{\min}(N, n, Y)$

Označme pro jednoduchost $G = (N \bmod |Y|)$ a $H = (Nm \bmod n)$. V práci (Somol a kol., 2008a) bylo ukázáno že

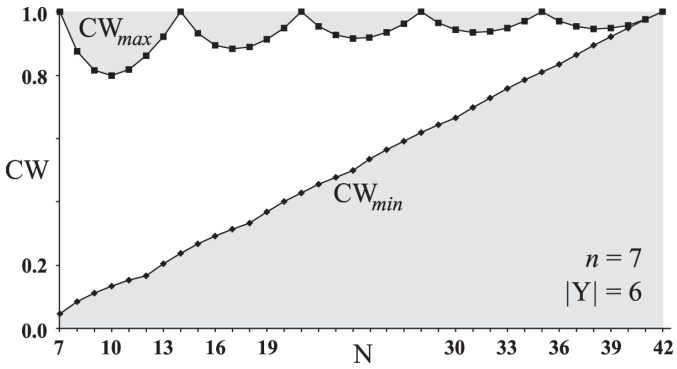
$$CW_{\min}(N,n,Y) = \frac{N^2 - |Y|(N-G) - G^2}{|Y| N(n-1)} \tag{13.35}$$

a

$$CW_{\max}(N,n) = \frac{H^2 + N(n-1) - Hn}{N(n-1)}. \tag{13.36}$$

Relativní vážená konzistence pak nabývá tvar

$$CW_{\mathrm{rel}}(\mathsf{S},Y) = \frac{|Y|\left(N - G + \sum_{f \in Y} F_f(F_f - 1)\right) - N^2 + G^2}{|Y|\left(H^2 + n(N-H) - G\right) - N^2 + G^2}. \tag{13.37}$$



Obr. 13.16 Ilustrace mezí míry CW.

Meze vážené konzistence $CW_{\max}(N, n)$ a $CW_{\min}(N, n, Y)$ jsou ilustrovány na obr. 13.16. Poznamenejme, že CW_{rel} může být citlivá na malé změny systému, když se N blíží k maximu (pro dané N, n a n).

Lze ukázat, že pro každé N, n reprezentující nějaký systém podmnožin S a pro dané Y platí, že $0 \le CW_{\mathrm{rel}}(\mathsf{S},Y) \le 1$ a pro odpovídající systémy $\mathsf{S}_{\min}$ a $\mathsf{S}_{\max}$ platí, že $CW_{\mathrm{rel}}(\mathsf{S}_{\min}) = 0$ a $CW_{\mathrm{rel}}(\mathsf{S}_{\max}) = 1$.

Míra (13.34) nevykazuje nežádoucí chování dávající větší hodnoty pro systémy s velikostí podmnožin blízkou |*Y*|, tj. je nezávislá na velikosti podmnožin příznaků vybíraných zkoumanými metodami FS při pevném *Y*. Můžeme říci, že tato míra charakterizuje pro dané S, *Y* relativní náhodnost složení množin v systému S na stupnici mezi maximální a minimální hodnotou vážené konsistence (13 32).

Koncepčně odlišná míra je definována v práci (Kalousis a kol., 2007). Je odvozena z *Tanimotova indexu* (*koeficientu*) definovaného jako podíl velikosti průniku a velikosti sjednocení podmnožin S_i a S_j (Duda a kol., 2000)

$$S_K(S_i, S_j) = \frac{|S_i \cap S_j|}{|S_i \cup S_j|}. \tag{13.38}$$

Definice 7. Průměrný Tanimotův index systému S je definován výrazem

$$ATI(\mathsf{S}) = \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} S_K(S_i, S_j). \tag{13.39}$$

ATI(S) je průměrná míra podobnosti přes všechny páry podmnožin příznaků v S. Nabývá hodnot z intervalu [0, 1], přičemž 0 indikuje prázdný průnik mezi všemi páry podmnožin S_i, S_j, a 1 indikuje, že všechny podmnožiny systému S jsou identické.

13.9.3 Experimenty s mírami stability

Pro ilustraci uváděných měr stability jsme provedli několik experimentů. Budeme porovnávat *d*-optimalizující metodu DOS a metody BIF, SFS a SFFS v *d*-optimalizující úpravě (úprava viz odst. 13.6.1). Pro odlišení od standardního *d*-parametrizovaného průběhu vyhledávání označíme tyto metody BIF*, SFS*, a SFFS*. Podobně jako v odst. 13.6.1 porovnání provedeme na mamogramových datech *wdbc* (30 dim., 2 třídy: 357 benigních a 212 maligních vzorcích) z databáze UCI Repository (Asuncion a kol., 2007) při použití tří různých kritérií. Budeme vyhodnocovat přesnost klasifikace: bayesovského klasifikátoru pro normální rozložení, klasifikátoru 3-NN (Dasarathy, 1991) a SVM (Support Vector Machine) s jádrem radiální bázové funkce, viz (Chang a kol., 2001). Pro každou kombinaci metoda–klasifikátor byl výběr příznaků opakován 1000krát na vždy jinak náhodně vybrané datové podmnožině odpovídající 80 % původních dat. Při každém opakování výběru příznaků byly 2/3 této datové podmnožiny použity k výběru příznaků (kritérium v podobě odhadu úspěšnosti klasifikace vyhodnocováno pomocí 10násobné krosvalidace) a zbývající 1/3 použita pro testování.

Výsledky shrnujeme v tabulce 13.6. Všechny míry, *CW*, CW_{rel} a *ATI*, indikují metodu BIF* jakožto obecně nejstabilnější metodu výběru příznaků, což potvrzuje závěry z článku (Kuncheva, 2007). Porovnáme-li mezi sebou ostatní zkoumané metody, nejlepší stabilitu i nejmenší velikost nalézaných podmnožin v daném experimentu vykazuje DOS. Podobným způsobem je možné zkoumat vhodnost konkrétních metod pro konkrétní úlohy. Poznamenejme, že CW_{rel} je zde jedinou mírou, která správně odhaluje náhodný výběr příznaků (v tabulce hodnoty blízké 0).

Tabulka 13.6 Experimentální vyhodnocení stability metod výběru příznaků typu pouzdro

		přesnost klasif.		velikost podmn.		*CW*	CW_{rel}	*ATI*	Doba výpočtu
kritérium	metoda	Průměr	*St.odch*	Průměr	*St.odch*				
bayesovský klasifikátor	náhodný výběr	0,908	*0,059*	14,90	*8,39*	0,500	0,008	0,296	00:00:14
	BIF*	0,948	*0,004*	27,15	*4,09*	0,927	0,244	0,862	00:04:57
	SFS*	0,963	*0,003*	11,95	*5,30*	0,506	0,181	0,332	01.02.04
	SFFS*	0,969	*0,003*	12,17	*4,66*	0,556	0,259	0,387	09:13:03
	DOS	0,973	*0,002*	8,85	*2,36*	0,584	0,419	0,429	12:49:59
3NN	náhodný výběr	0,935	*0,061*	14,9	*8,30*	0,501	0,009	0,297	00:00:45
	BIF*	0,970	*0,002*	24,78	*3,70*	0,912	0,513	0,840	00:38:39
	SFS*	0,976	*0,002*	15,45	*5,74*	0,584	0,148	0,401	07:27:39
	SFFS*	0,979	*0,002*	17,96	*5,67*	0,658	0,149	0,481	33:53:55
	DOS	0,980	*0,001*	13,27	*4,25*	0,565	0,227	0,393	116:47:
SVM	náhodný výběr	0,942	*0,059*	14,94	*8,58*	0,502	0,008	0,295	00:00:50
	BIF*	0,974	*0,003*	21,67	*2,71*	0,929	0,774	0,875	01:01:48
	SFS*	0,982	*0,002*	9,32	*4,12*	0,433	0,185	0,283	07:13:02
	SFFS*	0,983	*0,002*	10,82	*4,58*	0,472	0,179	0,310	30:28:02
	DOS	0,985	*0,001*	8,70	*3,42*	0,442	0,222	0,295	74:28:51

Velmi malé hodnoty CW_{rel} mohou indikovat hlubší problém daného procesu výběru příznaků. Je možné, že v dané úloze nejsou příznaky dostatečně rozlišitelné (a metody výběru příznaků tak generují více či méně náhodné podmnožiny), popř. dochází k výraznému přeučení (přílišnému přizpůsobení speciálním vlastnostem momentálně navzorkované trénovací datové podmnožiny). Dále poznamenejme, že malé hodnoty stability jsou často doprovázeny většími odchylkami ve velikosti podmnožin.

13.10 Shrnutí a další vývoj

Tato kapitola byla věnována problému výběru příznaků ve statistickém rozpoznávání. *Výběr příznaků* je jedním z nejčastěji používaných přístupů k redukci dimenzionality, jež je první ze dvou klíčových fází řešení problémů rozpoznávání. Druhou fází je vlastní klasifikace, prováděná na příznakovém podprostoru určeném ve fázi první. Výhodou *výběru příznaků* oproti *extrakci* je zachování významu původních měření (např. v medicíně je výhodné určit, která potenciálně drahá měření je možné pro účely automatické diagnostiky vynechat). Podali jsme zejména přehled dostupných vyhledávacích strategií a kritické srovnání jejich vlastností výhod i omezení zejména ve smyslu použitelnosti pro výběr příznaků. Dále jsme navrhli několik přehledových dělení známých metod na základě různých pohledů na cíle či průběh řešení problému. Současný stav problematiky můžeme shrnout takto:

Optimální metody výběru příznaků přicházejí v úvahu pouze pro problémy nepříliš velké dimenzionality. Není-li časově únosné provést úplné prohledání, potřebujeme navíc kritérium ohodnocení podmnožin splňující podmínku monotónnosti, aby bylo možné použít některý z algoritmů větví a mezí (Branch & Bound). Ačkoliv nejmodernější varianty tohoto algoritmu běží až o dva řády rychleji než varianta základní, zůstává řada principiálních omezení. Skutečná výkonnost algoritmů větví a mezí může při nepříznivých vlastnostech dat a kriteriální funkce vést k zdlouhavému výpočtu ne nutně rychlejšímu, než by bylo úplné prohledání. Běh těchto algoritmů má v principu exponenciální charakter a pro konkrétní problém není doposud předem možné odhadnout očekávanou dobu výpočtu. Ačkoliv pro některé problémy algoritmus nachází řešení velmi rychle,

reálně je nutné počítat s omezením na problémy dimenzionality do asi 40 až 50 při obvyklé době výpočtu hodin až dnů. Za nejmodernější algoritmy větví a mezí lze považovat algoritmy využívající samoučícího predikčního mechanismu. Jsou jimi tzv. *rychlý algoritmus větví a mezí FBB* (odst. 13.4.4) a *algoritmus větví a mezí s částečnou predikcí BBPP* (odst. 13.4.3).

Suboptimální metody jsou zřejmě nejčastěji používanými metodami výběru příznaků. Ačkoliv nezaručují nalezení optima vzhledem k zvolenému kritériu, často toto optimum nacházejí, nebo se k němu značně přibližují. Na rozdíl od optimálních metod bývají suboptimální metody tolerantní k nemonotónnímu chování kritéria a jsou proto použitelné pro přímou maximalizaci úspěšnosti klasifikátorů. Časová náročnost suboptimálních metod je obvykle polynomická na rozdíl od exponenciální náročnosti metod optimálních. Úpravou parametrů lze často volit mezi délkou výpočtu a očekávanou blízkostí řešení k optimu. Základní suboptimální metody jsou velmi jednoduché a snadno implementovatelné. Tradiční metody lze obvykle rozdělit na metody zdola–nahoru a shora–dolů podle převažujícího vývoje kardinality momentálně prohledávaných podmnožin. Nejpokročilejšími metodami této řady metod jsou hojně citované *plovoucí metody*, které v jednom průchodu nachází řešení ve všech dimenzích a obvykle výrazně překonávají metody dosavadní. Ještě širší možnosti nabízejí *oscilační metody*, které na rozdíl od ostatních metod zpracovávají v každém kroku podmnožiny požadované kardinality (odst. 13.5.6). Algoritmus *dynamického oscilačního vyhledávání* navíc optimalizuje velikost podmnožiny příznaků. Alternativně lze k výběru příznaků využít i různých randomizovaných algoritmů, např. algoritmů genetických (Mayer a kol., 2000) atd.

Koncepčně odlišné jsou metody výběru příznaků založené na modelování struktury dat za použití směsi hustot pravděpodobnosti speciálního typu. Tento přístup je vhodný pro situace, když máme k dispozici rozsáhlá trénovací data, o jejichž vlastnostech ale nic nevíme a hrozí tedy, že jde o data obtížně modelovatelná, např. multimodální. Jmenujme dvě varianty tohoto přístupu, tzv. *aproximační metodu*, vhodnou k reprezentaci dat, a *divergenční metodu*, vhodnou k separaci tříd. V obou případech výpočet zahrnuje konvergující iterativní výpočty parametrů směsi, přičemž výběr příznaků je integrální součástí výpočtu v každé iteraci.

Oblast redukce dimenzionality poskytuje široké možnosti dalšího vývoje. Je možné věnovat úsilí urychlování doposud příliš pomalých optimálních metod. Na druhé straně lze uvažovat o sofistikovanějších hledacích schématech zvyšujících šance, že nalezneme optima u metod suboptimálních. Neméně důležité je ovšem zkoumat *vlastnosti kriteriálních funkcí* a dále *klasifikátorů*, v jejichž lepší výkonnost a přesnost má redukce dimenzionality vyústit.

Při použití existujících nástrojů je navíc nutné brát v úvahu řadu potenciálních problémů. Podobně jako při konstrukci klasifikátorů musíme při výběru příznaků zabránit *přetrénování*, neboli přílišnému přizpůsobení se výsledku trénovací množiny (Raudys, 2006). Aktuálním tématem je též zkoumání *stability metod výběru příznaků*. Přetrénování, malá stabilita, popř. další komplikace, mohou – nejsou-li včas rozeznány – zásadně negativně ovlivnit přesnost výsledného rozhodovacího pravidla na nových datech. Zejména u vysokodimenzionálních problémů (kategorizace textů, vyhledávání genů, zpracování obrazových dat na úrovni pixelů) jsou obvykle používány pouze nejjednodušší metody výběru příznaků, a to nejen pro výpočetní složitost, ale také kvůli přílišnému nebezpečí přetrénování. Aktuální výsledky však naznačují praktickou možnost netriviálního výběru příznaků i v takto obtížných problémech (Somol a kol., 2011).

Množství metod a nástrojů je dnes již tak rozsáhlé, že je vhodné uvažovat o vytvoření podpůrných systémů usnadňujících uživateli orientaci a volbu postupu v konkrétních případech. První kroky byly učiněny v podobě vytvoření souborných knihoven relevantních metod (van der Heijden a kol., 2004), popř. kompaktnějších softwarových nástrojů (Somol a kol., 2002), (Pudil a kol., 2002). Tvorba přehledného sjednocujícího systému zůstává jedním ze směrů práce také našeho týmu. Pro získání aktuálních informací a volně dostupné rozsáhlé softwarové knihovny Feature Selection Toolbox odkazujeme na portál http://fst.utia.cz.

Literatura

Arauzo-Azofra A., Benítez J. M., Castro J. L.: C-focus:A continuous extension of focus. In: *Proc. 7th Online World Conf. on Soft Computing in Industrial Applications*, 2003, s. 225–232.

Asuncion A., Newman D.: UCI machine learning repository, 2007, http://www.ics.uci.edu/ mlearn/ mlrepository.html.

Boekee D. E., van der Lubbe J. C. A.: Some aspects of error bounds in feature selection, *Pattern Recognition*, 11, 1979, s. 353–360.

Bouguila N.: On multivariate binary data clustering and feature weighting, *Computational Statistics and Data Analysis*, 54, 2010, s. 120–134.

Bouguila N., Guebaly W. E.: Discrete data clustering using finite mixture models, *Pattern Recognition*, 42(1), 2009, s. 33–42.

Caruana R., Freitag D.: Greedy attribute selection. In: *Int. Conf. on Machine Learning ML–94*. Morgan Kaufmann, 1994, s. 28–36.

Das S.: Filters wrappers and a boosting-based hybrid for feature selection. In: *ICML '01: Proc. 18th Int. Conf. on Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisko 2001, s. 74–81.

Dasarathy B. V. (ed.): Nearest Neighbor Pattern Classification Techniques, *IEEE Computer Society Press*, 1991.

Dash M., Choi K., Scheuermann P., Liu H.: Feature selection for clustering – a filter solution. In: *ICDM '02: Proc. 2002 IEEE Int. Conf. on Data Mining*, volume 00, IEEE Computer Society, Washington 2002, 115 s.

Debuse J. C. W., Rayward-Smith V. J.: Feature subset selection within a simulated annealing data mining algorithm, *J. Intell. Inf. Syst.*, 9(1), 1997, s. 57–81.

Devijver P. A., Kittler J.: *Pattern Recognition: A Statistical Approach*. Prentice Hall, Englewood Cliffs, Londýn 1982.

Duda R. O., Hart P. E., Stork D. G.: *Pattern Classification*. (2. vyd.), Wiley-Interscience, 2000.

Dunne K., Cunningham P., Azuaje F.: *Solutions to Instability Problems with equential Wrapper based Approaches to Feature Selection*. Technical Report TCD-CS-2002-28, Department of Computer Science, Trinity College, Dublin 2002.

Foroutan I., Sklansky J.: Feature selection for automatic classification of nongaussian data, *IEEE Trans. on Systems Man and Cybernetics*, 17, 1987, s. 187–198.

Fukunaga K.: *Introduction to Statistical Pattern Recognition*. (2. vyd.), Academic Press Professional, Inc., San Diego 1990.

Gengler M., Coray G.: Aparallel best-first Branch&Bound algorithm and its axiomatization, *Parallel Algorithms Appl*, 2(1–2), 1994, s. 61–80.

Graham M. W., Miller D. J.: Unsupervised learning of parsimonious mixtures on large spaces with integrated feature and component selection, *IEEE Trans. on Signal Processing*, 54(4), 2006, s. 1289–1303.

Guyon I., Elisseeff A.: An introduction to variable and feature selection, *J. Mach. Learn. Res.*, 3, 2003, s. 1157–1182.

Hamamoto Y., Uchimura S., Matsuura Y., Kanaoka T., Tomita S.: Evaluation of the branch and bound algorithm for feature selection. *Pattern Recognition Letters*, 11(7), 1990, s. 453–456.

Hastie T., Tibshirani R.: Discriminant analysis by gaussian mixtures, *Journal of the Royal Statistical Society*, 58, 1996, s. 155–176.

Heijden van der F., Duin R. P., de Ridder D., Tax D. M.: *Classification Parameter Estimation and State Estimation: An Engineering Approach Using MATLAB*. Wiley, New York 2004.

Hussein F., Ward R., Kharma N.: Genetic algorithms for feature selection and weighting a review and study, *icdar* 00, 2001, 1240 s.

Chaikla N., Qi Y.: Genetic algorithms in feature selection. In: *IEEE Int. Conf. on Systems Man and Cybernetics*., vol. 5, IEEE Computer Society, Washington 1999, s. 538–540.

Chang C.-C., Lin C.-J.: *LIBSVM: a library for SVM*, 2001, http://www.csie.ntu.edu.tw/˜cjlin/libsvm.

Chen X. W.: An improved branch and bound algorithm for feature selection, *Pattern Recognition Letters*, 24(12), 2003, s. 1925–1933.

Iamnitchi A., Foster I.: A problem-specific fault-tolerance mechanism for asynchronous distributed systems. In: *2000 International Conference on Parallel Processing (ICPP'00)*, IEEE, Washington, Brusel, Tokio 2000, s. 4–14.

Jain A., Zongker D.: Feature selection: Evaluation application and small sample performance, *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(2), 1997, s. 153–158.

Jensen R.: *Performing Feature Selection with ACO*, vol. 34 of *Studies in Computational Intelligence*. Springer, Berlín, Heidelberg 2006, s. 45–73.

Kalousis A., Prados J., Hilario M.: Stability of feature selection algorithms: A study on high-dimensional spaces. *Knowledge and Information Systems*, 12(1), 2007, s. 95–116.

Kirkpatrick S., Gelatt C. D., Vecchi M. P.: Optimization by simulated annealing, *Science*, 220, 1983, s. 671–680.

Kohavi R., John G. H.: Wrappers for feature subset selection, *Artif. Intell.*, 97(1–2), 1997, s. 273–324.

Koller D., Sahami M.: Toward optimal feature selection. In: *13th Int. Conf. on Machine Learning*, 1996, s. 284–292.

Kononenko I.: Estimating attributes: Analysis and extensions of relief. In: *ECML-94: Proc. European Conf. on Machine Learning*. Springer-Verlag, New York, Inc., Secaucus, NJ, 1994, s. 171–182.

Korf K. E.: Artificial intelligence search algorithms. In: *Algorithms and Theory of Computation Handbook*. CRC Press, 1999.

Krishnapuram B., Hartemink A. J.: A bayesian approach to joint feature selection and classifier design, *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(9), 2004, s. 1105–1111.

Kříž ek P., Kittler J., Hlaváč V.: Improving stability of feature selection methods. In: *Proc. 12th Int. Conf. on Computer Analysis of Images and Patterns*, vol. LNCS 4673, Springer-Verlag, Berlín, Heidelberg 2007, s. 929–936.

Kudo M., Sklansky J.: Comparison of algorithms that select features for pattern classifiers, *Pattern Recognition*, 33(1), 2000, s. 25–41.

Kumar V., Kanal L. N.: A general branch and bound formulation for understanding and synthesizing and/or tree search procedures, *Artificial Intelligence*, 21(1–2), 1983, s. 179–198.

Kuncheva L. I.: A stability index for feature selection. In: *Proc. 25th IASTED International Multi-Conference AIAP'07*. ACTA Press, Anaheim 2007, s. 390–395.

Law M. H. C., Figueiredo M. A. T., Jain A.: Simultaneous feature selection and clustering using mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9), 2004, 1154–1166.

Lawler E. L., Wood D. E.: Branch-and-bound methods: A survey, *Operations Research*, 14, 1966, s. 699–719.

Liu H., Setiono R.: Scalable feature selection for large sized databases. In: *Proceedings of the Fourth World Congress on Expert Systems*, Morgan Kaufmann, 1998, s. 521–528.

Liu H., Yu L.: Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, 17(4), 2005, s. 491–502.

Mayer H. A., Somol P., Huber R., Pudil P.: Improving statistical measures of feature subsets by conventional and evolutionary approaches. In: *Proceedings of the Joint IAPR International Workshops on Advances in Pattern Recognition*. Springer-Verlag, Londýn 2000, s. 77–86.

McKenzie P., Alder M.: Initializing the EM algorithm for use in Gaussian mixture modelling. *Machine Intelligence and Pattern Recognition*, 16, 1994, s. 91.

McLachlan G. J.: *Discriminant analysis and statistical pattern recognition*. Wiley-IEEE, 2004.

Miller D. J., Browning J.: A mixture model and EM-based algorithm for class discovery robust classification and outlier rejection in mixed labeled/unlabeled data sets, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(11), 2003, s. 1468–1483.

Mitschele-Thiel A.: Integrated system development with the DSPL programming environment. In: Joubert G. R., Trystram D., Peters F. J., Evans D. J. (eds): *Parallel Computing: Trends and Applications PARCO'93*, Elsevier, Grenoble 1994, s. 635–638.

Nakariyakul S., Casasent D. P.: Adaptive branch and bound algorithm for selecting optimal features, *Pattern Recogn. Lett.*, 28(12), 2007, s. 1415–1427.

Nakariyakul S. & Casasent D. P.: An improvement on floating search algorithms for feature subset selection, *Pattern Recognition*, 42(9), 2009, s. 1932–1940.

Narendra P. M., Fukunaga K.: A branch and bound algorithm for feature subset selection, *IEEE Trans. Computers*, 26(9), 1977, s. 917–922.

Nilsson N.: *Problem Solving Methods in Artificial Intelligence*. McGraw-Hill, New York 1971.

Nilsson N. J.: *Artificial Intelligence: A New Synthesis*. Morgan Kaufmann Publishers, San Francisko 1998.

Novovičová J., Pudil P., Kittler J.: Divergence based feature selection for multimodal class densities, *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(2), 1996, s. 218–223.

Novovičová J., Somol P., Pudil P.: Oscillating feature subset search algorithm for text categorization. In: *Structural Syntactic and Statistical Pattern Recognition*, vol. LNCS 4109, Springer-Verlag, Berlín, Heidelberg 2006, s. 578–587.

Novovičová J., Pudil P.: Feature selection and classification by modified model with latent structure. In: Karný M., Warwick K., Kurková V. (eds): *Dealing with Complexity: A Neural Networks Approach*. Springer Verlag, 1998, s. 126–140.

Palm H. C.: A new method for generating statistical classifiers assuming linear mixtures of gaussian densities. In: *ICPR*, 1994, s. B:483–486.

Pudil P., Novovičová J., Choakjarernwanit N., Kittler J.: Feature selection based on the approximation of class densities by finite mixtures of special type, *Pattern Recognition*, 28(9), 1995, s. 1389–1398.

Pudil P., Novovičová J., Kittler J.: Floating search methods in feature selection, *Pattern Recognition Letters*, 15(11), 1994a, s. 1119–1125.

Pudil P., Novovičová J., Somol P.: Feature selection toolbox software package, *Pattern Recognition Letters*, 23(4), 2002, s. 487–492.

Pudil P., Novovičová J., Kittler J.: Simultaneous learning of decision rules and important attributes for classification problems in image analysis, *Image and Vision Computing*, 12, 1994b, s. 193–198.

Raudys Š. J.: Feature over-selection. In: *Structural Syntactic and Statistical Pattern Recognition*, vol.LNCS 4109, Springer-Verlag, Berlín, Heidelberg 2006, s. 622–631.

Redner R. A., Walker H. F.: Mixture densities maximum likelihood and the EM algorithm, *SIAM Review*, 26(2), 1984, s. 195–239.

Reunanen J.: Overfitting in making comparisons between variable selection methods, *J. Mach. Learn. Res.*, 3, 2003, s. 1371–1382.

Ripley B. D. (ed.): *Pattern Recognition and Neural Networks*. 8. vyd., Cambridge University Press, 2005.

Saeys Y., Inaki I. I., Larranga P. L.: A review of feature selection techniques in bioinformatics, *Bioinformatics*, 23(19), 2007, s. 2507–2517.

Sebastiani F. Machine learning in automated text categorization, *ACM Computing Surveys*, 34(1), 2002, s. 1–47.

Sebban M., Nock R.: A hybrid filter/wrapper approach of feature selection using information theory, *Pattern Recognition*, 35, 2002, s. 835–846.

Skurichina M.: *Stabilizing Weak Classifiers*. PhD thesis, Pattern Recognition Group, Delft University of Technology, Netherlands, 2001.

Somol P., Novovičová J., Pudil P.: Flexible-hybrid sequential floating search in statistical feature selection. In: *Structural Syntactic and Statistical Pattern Recognition*, vol. LNCS 4109, Springer-Verlag, Berlín, Heidelberg 2006, s. 632–639.

Somol P. & Novovičová J. (2008a). Evaluating the stability of feature selectors that optimize feature subset cardinality. In *Structural Syntactic and Statistical Pattern Recognition* vol. LNCS 5342 pp. 956–966.

Somol P., Novovičová J., Grim J., Pudil P.: Dynamic oscillating search algorithms for feature selection. In: *ICPR 2008*, IEEE Computer Society, Los Alamitos, CA, 2008b.

Somol P., Pudil P.: Oscillating search algorithms for feature selection. In: *ICPR 2000*, vol. 02, IEEE Computer Society, Los Alamitos, CA, 2000, s. 406–409.

Somol P., Pudil P.: Feature selection toolbox, *Pattern Recognition*, 35(12), 2002, s. 2749–2759.

Somol P., Pudil P., Kittler J.: Fast branch&bound algorithms for optimal feature selection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(7), 2004, s. 900–912.

Somol P., Pudil P., Novovičová J., & Paclík P.: Adaptive floating search methods in feature selection, *Pattern Recognition Letters*, 20(11–13), 1999, s. 1157–1163.

Somol P., Grim J., Pudil P.: Fast Dependency-Aware Feature Selection in Very-High-Dimensional Pattern Recognition. In: *IEEE SMC*, 2011, s. 502–509.

Sun Y.: Iterative relief for feature weighting: Algorithms theories and applications, *IEEE Trans. Pattern Anal. Mach. Intell*, 29(6), 2007, s. 1035–1051.

Theodoridis S., Koutroumbas K. *Pattern Recognition*. 3. vyd., Academic Press, 2006.

Webb A. R.: *Statistical Pattern Recognition*, 2. vyd., John Wiley and Sons Ltd., 2002.

Webb G. I.: OPUS: An efficient admissible algorithm for unordered search, *Journal of Artificial Intelligence Research*, 3, 1995, s. 431–465.

Whitney A. W.: A direct method of nonparametric measurement selection, *IEEE Trans. Comput.*, 20(9), 1971 s. 1100–1103.

Xing E. P.: *Feature Selection in Microarray Analysis*. Springer, 2003, s. 110–129.

Xu C., Tschöke S., Monien B.: Performance evaluation of load distribution strategies in parallel branch and bound computations. In: *Proceedings of the 7th IEEE Symposium on Parallel and Distributed Processing SPDP'95*, 1995, s. 402–405.

Yang M., Das C.: A parallel optimal branch-and-bound algorithm for min-based multiprocessors. In: *Proceedings of the IEEE Int. Conf. on Parallel Processing*, 1999, s. 112–119.

Yang Y., Pedersen J. O.: A comparative study on feature selection in text categorization. In: *ICML '97: Proc. 14th Int. Conf. on Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisko 1997, s. 412–420.

Yu B., Yuan B. Z.: A more efficient branch and bound algorithm for feature selection, *Pattern Recognition*, 26(6), 1993, s. 883–889.

Yu L., Liu H.: Feature selection for high-dimensional data: A fast correlation based filter solution. In: *ICML-03: Proc. 20th Int. Conf. on Machine Learning*, vol. 20, Morgan Kaufmann Publishers Inc., Washington 2003, s. 856–863.

Zhang H., Sun G.: Feature selection using tabu search method, *Pattern Recognition*, 35, 2002, s. 701–711.