



národní
úložiště
šedé
literatury

Porovnání optimalizačních metod pro změkčování rozhodovacího stromu

Dvořák, Jakub
2009

Dostupný z <http://www.nusl.cz/ntk/nusl-40435>

Dílo je chráněno podle autorského zákona č. 121/2000 Sb.

Tento dokument byl stažen z Národního úložiště šedé literatury (NUŠL).

Datum stažení: 11.05.2024

Další dokumenty můžete najít prostřednictvím vyhledávacího rozhraní nusl.cz .

Porovnání optimalizačních metod pro změkčování rozhodovacího stromu

doktorand:

MGR. JAKUB DVOŘÁK

Ústav informatiky AV ČR, v. v. i.
Pod Vodárenskou věží 2

182 07 Praha 8

dvorak@cs.cas.cz

školitel:

RNDR. PETR SAVICKÝ, CSc.

Ústav informatiky AV ČR, v. v. i.
Pod Vodárenskou věží 2

182 07 Praha 8

savicky@cs.cas.cz

obor studia:
Teoretická informatika

Tento výzkum byl podporován institucionálním výzkumným záměrem AV0Z10300504 a také projektem T100300517 programu „Informační společnost“ AV ČR.

Abstrakt

Zkoumáme metody změkčování rozhodovacích stromů, které vycházejí z hotového rozhodovacího stromu získaného metodou CART a při zachování jeho struktury hledají změkčení tak, že optimalizují kvalitu klasifikátoru na trénovací množině. Představené metody používají dvě různé míry kvality klasifikátoru. Jednou z nich je součet jistým způsobem transformované chyby na jednotlivých vzorech trénovací množiny, druhou je plocha pod ROC křivkou (AUC). K hledání co nejlepšího změkčení je použita randomizovaná strategie iterované optimalizace, která v každém cyklu modifikuje pouze několik parametrů. V rámci této strategie využíváme jako optimalizační metody simulované žhání a simplexový algoritmus pro minimalizaci — Nelder-Mead. Jako ukončující kritérium pro iterační proces změkčování za účelem porovnání metod používáme dosažení limitu reálného času výpočtu. V experimentech s daty „Magic Telescope“ při porovnání podle AUC se ukazuje jako nejlepší optimalizace AUC pomocí metody Nelder-Mead.

1. Úvod

Změkčování rozhodovacích stromů je cesta ke zlepšení kvality predikce klasifikátoru na prostoru vzorů s reálnými atributy. Jestliže výstupem klasifikátoru je reálné číslo, změkčené stromy umožňují, aby tento výstup byl spojitou funkcí atributů. V nezměkčeném rozhodovacím stromu jsou ve vnitřních uzlech podmínky, které na základě předloženého vzoru rozhodují, zda v prohledávání pokračovat v levém nebo pravém potomkovi daného uzlu. Když prohledávání dosáhne listu, je z něj zjištěn výsledek klasifikace.

Změkčování je založeno na tom, že rozhodovací podmínky ve vnitřních uzlech („splitech“) jsou nahrazeny pravidlem výpočtu poměru, v jakém jsou zkombinovány výsledky levého a pravého podstromu.

Zde budeme vycházet z nezměkčeného klasického rozhodovacího stromu pro klasifikaci získaného metodou CART [2], jehož rozhodovací podmínky budeme následně změkčovat. Pro takové změkčování jakožto postprocessing budeme s použitím trénovací množiny optimalizovat kvalitu klasifikátoru. Zaměřujeme se na klasifikaci do dvou tříd nazývaných „signal“ a „background“, výstupem klasifikátoru je pro každý předložený vzor reálné číslo v intervalu $[0, 1]$, které je odhadem pravděpodobnosti, že vzor patří do třídy „signal“.

Metoda změkčování, která byla v našich předchozích výzkumech nejuspěšnější [3], hledala co nejlepší vektor parametrů změkčení tak, že opakovaně vybírala několik parametrů, a na této podmnožině parametrů používala simulované žhání k optimalizaci cílové funkce, ostatní parametry zůstávaly zatím zafixovány. Cílová funkce v této metodě byla počítána tak, že pro vzory z trénovací množiny byla vypočtena klasifikace stromem změkčeným s danými parametry, pro každý vzor byla absolutní hodnota rozdílu získané klasifikace od správné hodnoty (tj. 0 nebo 1) transformována exponenciální funkcí a výsledky sečteny přes celou trénovací množinu. Tato metoda však konvergovala velmi pomalu, získání dostatečně kvalitního změkčeného stromu trvalo několik hodin.

Následujícím cílem bylo nalezení metody, jež umožní získání alespoň stejně dobrého změkčeného stromu v rozumném čase. Zde budeme porovnávat změkčující metody, pro něž jsme zvolili jako ukončující kritérium cyklů optimalizačních iterací vyčerpání časového limitu.

Získané klasifikátory budeme porovnávat podle plochy pod ROC křivkou (Area Under Curve, AUC) [4] naměřené na testovacích datech. AUC je standardní míra kvality klasifikátoru. Hodnota AUC leží v intervalu $[0, 1]$, čím je vyšší, tím je klasifikátor lepší. AUC pro náhodný klasifikátor je $1/2$. Interpretace je následující: Vybereme-li náhodně rovnoměrně jeden pozitivní případ a jeden negativní případ, potom AUC je pravděpodobnost, že klasifikátor pro vybraný pozitivní případ vydá vyšší výstupní hodnotu, než pro vybraný negativní případ.

Jedna ze zde zkoumaných metod ještě aplikuje myšlenku výše zmíněné cílové funkce. Další metody jako cílovou funkci používají odhad AUC na základě trénovacích dat. Tuto funkci maximalizují v jednom případě opět iterovaným simulovaným žháním a v druhém případě je v rámci iterační strategie použita metoda „Nelder-Mead“.

2. Změkčování rozhodovacího stromu

Mějme (nezměkčený) rozhodovací strom, pro klasifikaci vzorů s m reálnými atributy. Označme s počet vnitřních uzlů (splitů), tedy včetně listů má strom $2s + 1$ uzlů. Uzly stromu označujeme v_j , $j = 1, \dots, 2s + 1$, předpokládejme, že splity mají indexy $1, \dots, s$ a listy $s + 1, \dots, 2s + 1$. Každému splitu v_j je přiřazen vektor $\mathbf{w}_j = (w_{j,1}, \dots, w_{j,m})$ a reálné číslo c_j , každému listu v_j je přiřazeno reálné číslo r_j . Metoda CART používá jako hodnotu r_j relativní četnost signal případů v listu v_j , takže

$$0 \leq r_j \leq 1, \quad r = s + 1, \dots, 2s + 1$$

Klasifikace vstupního vzoru $\mathbf{x} = (x_1, \dots, x_m)$ probíhá tak, že od kořene stromu jakožto výchozího aktuálního uzlu se provádí následující proces: Když v_j je vnitřní uzel, provede se test

$$\mathbf{w}_j \mathbf{x} \leq c_j \quad (1)$$

a pokud je nerovnost (1) splněna, aktuálním uzlem se stává levý potomek uzlu v_j , jinak pravý potomek. Je-li aktuální uzel v_j list, potom výstupem klasifikátoru je hodnota r_j .

Tímto způsobem získáme jako výstup klasifikátoru reálné číslo (jedno z čísel r_j přiřazených některému listu). Pokud je hodnota výstupu klasifikátoru větší než zvolený práh, zařazujeme předložený vzor do třídy „signal“, jinak do třídy „background“.

V experimentech používáme stromy, kde vektory \mathbf{w}_j obsahují právě jednu jedničku a jinak samé nuly, tzn.

výraz $\mathbf{w}_j \mathbf{x}$ má hodnotu x_i , kde i je takový index, že $w_{j,i} = 1$, čili vektor \mathbf{w}_j vyjadřuje výběr jednoho atributu předloženého vzoru. To opět odpovídá klasické metodě CART, i když existuje i modifikace, která ve splittech pro porovnávání používá obecné netriviální lineární kombinace atributů.

Rozhodovací strom rozděluje prostor vzorů na oblasti, v našem případě s právě jednou jedničkou ve vektoru \mathbf{w}_j se jedná o hyperkvádry se stěnami kolmými na osy souřadné soustavy určené atributy. V každém hyperkvádru je výstup klasifikátoru pro všechny body stejný. Změkčování splitů ve stromu se projeví tím, že skokové přechody výstupu klasifikátoru na hranicích hyperkvádrů se změní na spojitě.

Při změkčování každému splitu v_j , $j = 1, \dots, s$, přiřadíme parametry změkčení

$$a_j, b_j \geq 0 \quad (2)$$

Potom definujeme změkčující funkci (viz obrázek 1) příslušnou uzlu v_j , parametrizovanou čísly a_j, b_j, c_j a vektorem \mathbf{w}_j :

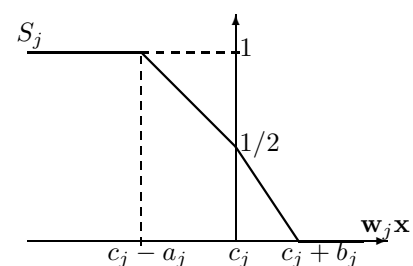
$$S_j(\mathbf{x}) = \sigma_{a_j, b_j}(\mathbf{w}_j \mathbf{x} - c_j)$$

kde $\sigma_{a,b}$ lineárně interpoluje body uvedené v tabulce:

t	$-\infty$	$-a$	0	b	∞
$\sigma_{a,b}(t)$	1	1	1/2	0	0

Pro případ, že pro nějaké $j \leq s$ je $a_j = 0$, nebo $b_j = 0$ je potřeba dodefinovat

$$\begin{aligned} \sigma_{0,b}(0) &= 1 && \text{pro každé } b \geq 0 \\ \sigma_{a,0}(0) &= 1/2 && \text{když } a > 0 \end{aligned}$$



Obrázek 1: Změkčující funkce

Pro vstupní vzor \mathbf{x} a uzel v_j změkčeného stromu definujeme výstup uzlu $v_j(\mathbf{x})$ následující rekurzí: Je-li v_j list, je jeho výstupem r_j . Pro vnitřní uzel v_j označme v_j^L resp. v_j^R jeho levého resp. pravého potomka. Potom

$$v_j(\mathbf{x}) = S_j(\mathbf{x})v_j^L(\mathbf{x}) + (1 - S_j(\mathbf{x}))v_j^R(\mathbf{x})$$

Výstupem klasifikátoru je výstup kořene stromu. I u změkčeného stromu pro finální klasifikaci použijeme porovnání výstupu klasifikátoru se zvoleným prahem.

Jestliže $a_j = 0$ a $b_j = 0$ pro všechna $j \leq s$, potom výstup změkčeného stromu je pro každý vstupní vzor roven výstupu původního nezměkčeného stromu.

Úlohou změkčování daného stromu je hledání parametrů a_j, b_j pro všechny vnitřní uzly $j = 1, \dots, s$. Vektor všech parametrů a_j, b_j budeme také označovat \mathbf{p} . Změkčený strom s parametry změkčení \mathbf{p} budeme označovat $T^{(\mathbf{p})}$ přičemž $T^{(\mathbf{p})}(\mathbf{x})$ znamená výstup tohoto klasifikátoru pro vzor \mathbf{x} .

3. Iterování optimalizace na podmnožinách parametrů

V této sekci popíšeme strategii hledání parametrů minimalizujících cílovou funkci, která používá opakovaně optimalizační metodu, vždy pouze na podmnožině parametrů a ostatní parametry zůstávají konstantní. Tedy jednotlivé optimalizační běhy řeší úlohu nižší dimenze.

K tomuto účelu zavedeme následující značení: Necht $Q \subseteq \{1, \dots, 2s\}$ a $\mathbf{z} \in \mathbb{R}^{2s}$. \mathbb{R}^Q bude označovat množinu vektorů $\{x_i\}_{i \in Q}$, tzn. vektorů z $\mathbb{R}^{|Q|}$, jejichž složky jsou indexovány prvky Q místo čísel $1, \dots, |Q|$. Máme-li cílovou funkci změkčování $f(\mathbf{p})$ definovanou na \mathbb{R}^{2s} , potom pro $\mathbf{z} = (z_1, \dots, z_{2s})$ necht $f[Q, \mathbf{z}] : \mathbb{R}^Q \rightarrow \mathbb{R}$ je funkce definovaná na každém $\mathbf{x} \in \mathbb{R}^Q$ tak, že $f[Q, \mathbf{z}](\mathbf{x}) = f(\mathbf{y})$, kde

$$y_i = \begin{cases} x_i & \text{když } i \in Q \\ z_i & \text{jinak} \end{cases}$$

Iterační strategie opakovaně aplikuje vybranou optimalizační metodu, cílovou funkcí je $f[Q, \mathbf{p}_0](\mathbf{p}')$, která má $|Q|$ argumentů, kde \mathbf{p}_0 je výsledek předchozího volání, nebo v případě první iterace iniciační hodnota změkčování. Restrikce \mathbf{p}_0 na vybranou množinu indexů Q je iniciační hodnota pro \mathbf{p}' v aktuální iteraci.

Výběr podmnožiny Q parametrů v každém cyklu je randomizovaný a založený na struktuře stromu. Nejprve je náhodně zvolen jeden z parametrů změkčení jako kořenový parametr. Jestliže kořenový parametr je a_j pro nějaké j , potom v_j^X označíme v_j^L , tedy levého potomka uzlu v_j . Jestliže kořenový parametr je b_j pro nějaké j , potom v_j^X rozumíme v_j^R — pravého potomka uzlu v_j .

Kořenový parametr volíme tak, aby v_j ani v_j^X nebylo listem stromu, mezi takovými parametry je náhodný výběr kořenového parametru rovnoměrně rozdělen. Do výsledné podmnožiny parametrů pro optimalizaci Q zahrneme kořenový parametr, oba parametry příslušné uzlu v_j^X a všechny parametry příslušné přímým potomkům uzlu v_j^X . Množina Q tak může mít 7 prvků, ale protože jeden nebo oba z přímých potomků uzlu v_j^X mohou být listy, může mít tato podmnožina také 5 nebo 3 prvky.

4. Optimalizační metody

Porovnávané metody změkčování jsou založeny na dvou standardních optimalizačních metodách — simulovaném žhání a simplexovém algoritmu pro minimalizaci „Nelder-Mead“ [5]. Obě tyto metody hledají optimum iterativně pouze na základě funkčních hodnot cílové funkce, tedy bez potřeby výpočtu diferenciálu, takže jsou použitelné i na optimalizace nespojitých funkcí.

Protože použité implementace těchto optimalizačních metod neumožňovaly omezení definičního oboru cílové funkce, ale přitom na parametry změkčení stromu jsou kladeny podmínky (2), je cílová funkce dodefinována tak, aby pro vstupní hodnoty porušující (2), generovala vysokou hodnotu (viz níže) a tím uměle nutila optimalizační metody se této oblasti vyvarovat.

V rámci strategie iterované optimalizace popsané v předešlé sekci jsou metody použity následujícím způsobem: Simulované žhání — v každé iteraci se provádí 100 kroků metody, nový kandidátský bod se generuje na základě Gauss-Markovova kernelu, iniciační teplota je nastavena na hodnotu 10 a update teploty se provádí po každém kroku. Nelder-Mead — v každé iteraci je provedeno 30 kroků metody.

Pro určení škály a iniciační hodnoty pro optimalizační metody používáme vzdálenost splitu od hranice hyperkvádrů. Tyto hodnoty definujeme takto: Nejprve celý prostor vzorů ve směrech všech atributů omezíme nejzazšími trénovacími vzory, tak získáme základní hyperkvádr. Když v uzlu v_j podmínka (1) rozděluje hyperkvádr vyšší úrovně, který je v testované proměnné x_i omezen hodnotami $z_{j,1}, z_{j,2}$, kde $z_{j,1} < c_j < z_{j,2}$, potom za vzdálenost od hranice hyperkvádrů příslušnou pro parametr a_j resp. b_j považujeme

$$h_j^a = c_j - z_{j,1}; \quad h_j^b = z_{j,2} - c_j$$

metoda	A	B	C	D
iniciální hodnota	$a_j^0 = 0; \quad b_j^0 = 0$	$a_j^0 = 1/4 h_j^a; \quad b_j^0 = 1/4 h_j^b$		
ukončující kritérium	50 iterací po sobě bez zlepšení	vyčerpaný časový limit 5 minut		
cílová funkce	φ_A	φ_B	AUC	
optimalizační metoda	iterované simulované žhání			iter. Nelder-Mead
škála	$(h_j^a, h_j^b), j = 1, \dots, s$			$(1/16 h_j^a, 1/16 h_j^b)$ $j = 1, \dots, s$

Tabulka 1: Přehled metod změkčování

Základní charakteristiky zkoumaných metod ukazuje tabulka 1. Písmeno A označuje metodu z [3]. Tato metoda používá cílovou funkci definovanou pro legální parametry změkčení \mathbf{p} (viz (2)) jako

$$\varphi_A(\mathbf{p}) = \sum_{i=1}^n \exp\left(4 \left(\left| T^{(\mathbf{p})}(\mathbf{x}_i) - y_i \right| - 1 \right)\right)$$

kde \mathbf{x}_i , $i = 1, \dots, n$ jsou prvky testovací množiny a y_i jsou jim příslušné klasifikace, tedy hodnoty 0 nebo 1 pro background resp. signal případy. Pro nelegální parametry je $\varphi_A(\mathbf{p}) = n + 1$.

Pod písmenem B uvádíme metodu, která je založena na stejném základním principu, jako metoda A, ale obsahuje několik zlepšení nalezených od publikování [3], zejména nenulový inicializační vektor parametrů, normalizaci funkční hodnoty a výstupní hodnotu při nelegálních parametrech, jež roste se vzdáleností hodnoty každého nelegálního parametru od legálních hodnot. Pro tuto metodu již používáme jako ukončující kritérium časový limit. To nám umožní relevantnější porovnání myšlenky metody A s ostatními metodami.

Cílová funkce pro metodu B při legálních parametrech je

$$\varphi_B(\mathbf{p}) = \frac{1}{n} \varphi_A(\mathbf{p})$$

Pro nelegální parametry, tzn. pokud některá ze složek vektoru \mathbf{p} je záporná, definujeme

$$\mu(\mathbf{p}) = 1 + \sum_{i=1}^{2s} \max(0, -p_i)$$

$$\varphi_B(\mathbf{p}) = \mu(\mathbf{p})$$

Pro další metody je v tabulce 1 uvedena jako cílová funkce „AUC“, což přesněji znamená, že se

¹<http://www.gnu.org/software/gsl/>

minimalizuje funkce, která pro legální parametry \mathbf{p} má hodnotu záporně vzaté AUC pro $T^{(\mathbf{p})}$ naměřené na trénovací množině, pro nelegální parametry má hodnotu $\mu(\mathbf{p})$.

5. Implementace

Protože jsme použili v ukončujícím kritériu optimalizace reálný čas, je důležitá implementace experimentů. Základním frameworkem byl systém R [6], který zahrnuje interpret jazyka a rozšiřitelný systém balíčků, díky němuž mohou být jednotlivé metody naprogramovány např. v jazyce C a integrovány pomocí zkompileované sdílené knihovny.

V jazyce R byla naprogramována nejvyšší úroveň sestavení experimentů s využitím následujících komponent:

- Klasifikace množiny vzorů změkčeným stromem, byla implementována v jazyce C.
- Výpočet AUC — byl v jazyce C.
- Metoda simulovaného žhání byla v jazyce C, jednalo se o mírně upravenou implementaci, jež je součástí systému R.
- Metoda Nelder-Mead byla integrována implementace z knihovny GNU Scientific Library¹ pomocí R package „gsl“.
- Strategie iterované optimalizace na podmnožinách množiny parametrů byla implementována v jazyce R.

Výpočty běžely na procesoru Intel[®] Xeon[™] CPU 2.80GHz, v systému se 4GB operační paměti.

6. Experimenty

V experimentech byla použita data „Magic Telescope”², která jsou zkoumána také v [3]. Problematikou klasifikace těchto dat se více zabývá [1]. Data mají 10 reálných atributů, obsahují přibližně 65 % signal případů.

Trénovací množina obsahující 12680 vzorů, byla rozdělena na dvě části v poměru velikostí 2:1, první část byla použita pro růst stromu a druhá část jako validační množina pro prořezávání. Stromy byly vytvořeny metodou CART, nastavením různých hodnot parametrů prořezávání byla získána sekvence stromů různých velikostí. V této sekvenci byl na počátku největší strom a každý další vzniknul prořezáním předchozího, tedy byl jeho podstromem. Pro změkčování byly použity z celé sekvence pouze ty stromy, které nebyly větší, než strom, který měl na validační množině nejmenší chybu. Jako data sloužící k výpočtu cílové funkce při změkčování byla potom použita celá trénovací množina.

Z důvodu velké časové náročnosti metody A byl vygenerován pouze malý počet stromů změkčených touto metodou, změkčovány byly stromy z počátku sekvence prořezávání, čili největší, tedy nejpřesnější stromy (podrobnosti viz [3]). Pro porovnání jsme pro každý strom ze sekvence spočetli tolik změkčení každou z metod B, C, D, kolik bylo k dispozici změkčení metodou A.

Na základě testovací množiny obsahující 6340 vzorů byla vypočtena hodnota AUC pro každý takto získaný klasifikátor. Pro každou z metod jsme vypočetli průměrnou a maximální hodnotu AUC ze všech změkčených stromů.

Celý popsaný postup byl opakován desetkrát s tím, že pro každý experiment byla dostupná data nově rozdělena na trénovací a testovací množinu. Díky odlišným trénovacím množinám byly v různých experimentech odlišné primární stromy, které byly základem pro prořezávání. Tabulka 2 ukazuje počet vnitřních uzlů největšího stromu ze sekvence použitého pro změkčování a jeho hodnotu AUC naměřenou na testovacích datech.

Průměrné hodnoty AUC stromů změkčených jednotlivými metodami porovnává tabulka 3, maximální hodnoty tabulka 4.

Metody B a C mají výsledky obecně horší, než metoda A. Výsledky metod A a D porovnává tabulka 5. Průměrné hodnoty AUC metody D jsou pouze ve dvou případech z 10 nepatrně horší, než u metody A, maximální hodnoty dokonce jen v jednom případě z 10.

²<http://www.magic.mppmu.mpg.de>

	počet splitů	AUC		počet splitů	AUC
1	45	0.887254	6	69	0.886673
2	49	0.882268	7	38	0.881902
3	69	0.886131	8	52	0.880057
4	64	0.892350	9	75	0.893006
5	43	0.894513	10	56	0.885681

Tabulka 2: Vlastnosti nezměkčených stromů.

	A	B	C	D
1	0.909050	0.903945	0.904057	0.912239
2	0.907109	0.896344	0.898889	0.908804
3	0.914037	0.902580	0.906368	0.914319
4	0.913617	0.903198	0.903999	0.915339
5	0.913058	0.905683	0.907888	0.917001
6	0.909587	0.897866	0.899282	0.909323
7	0.908522	0.901193	0.904355	0.909901
8	0.907109	0.897537	0.899769	0.908703
9	0.916947	0.906992	0.909030	0.917126
10	0.913255	0.903054	0.904603	0.912520

Tabulka 3: Průměrné hodnoty AUC.

	A	B	C	D
1	0.913832	0.907214	0.907320	0.913885
2	0.909889	0.898939	0.904546	0.910490
3	0.917025	0.905709	0.910231	0.917092
4	0.918478	0.908102	0.908378	0.918705
5	0.916306	0.909574	0.911750	0.919407
6	0.913164	0.903933	0.903855	0.911248
7	0.910700	0.905010	0.911652	0.914576
8	0.909786	0.901399	0.903703	0.911067
9	0.919530	0.911230	0.915396	0.920273
10	0.915340	0.907896	0.910725	0.916069

Tabulka 4: Maximální hodnoty AUC.

	$\varnothing D / \varnothing A$	max D / max A
1	1.0035087	1.0000580
2	1.0018689	1.0006610
3	1.0003088	1.0000739
4	1.0018849	1.0002472
5	1.0043183	1.0033848
6	0.9997104	0.9979017
7	1.0015178	1.0042566
8	1.0017566	1.0014079
9	1.0001951	1.0008084
10	0.9991960	1.0007963

Tabulka 5: Poměry hodnot AUC metody D a A.

7. Závěr

Porovnali jsme 4 metody pro změkčování rozhodovacího stromu založené na optimalizaci kvality klasifikátoru na trénovací množině. Cílem bylo dosáhnout v rozumném čase alespoň srovnatelných výsledků získaného klasifikátoru, jaké dávala metoda založená na iterovaném simulovaném žhání, která používala jako cílovou funkci φ_A , tedy součet exponenciální funkcí transformovaných vzdáleností výstupu klasifikátoru od správné klasifikace.

Představili jsme metodu podobnou — také založenou na iterovaném simulovaném žhání a uvedené cílové funkci, ale se zlepšeními v oblasti inicializace, řešení ilegálních hodnot a normalizace funkční hodnoty. Tato zlepšení nevedla k tomu, že by metoda v daném pětiminutovém časovém limitu dosahovala dostatečně kvalitních změkčení.

V experimentech se ukázalo, že lepší cílovou funkcí je plocha pod ROC křivkou (AUC). Pro tuto cílovou funkci jsme použili jako optimalizační strategie opět iterované simulované žhání a také iterovaný simplexový algoritmus (Nelder-Mead). Poslední z uvedených metod dosáhla na datové množině „Magic Telescope” při výpočtu omezeném v čase 5ti minut výsledků srovnatelných s těmi, které původní metoda počítala několik hodin.

Literatura

- [1] R.K. Bock, A. Chilingarian, M. Gaug, F. Hakl, T. Hengstebeck, M. Jiřina, J. Klaschka, E. Kotrč, P. Savický, S. Towers, and A. Vaicilius, “Methods for multidimensional event classification: a case study using images from a Cherenkov gamma-ray telescope.” *Nucl. Instr. Meth.*, A 516, pp. 511–528, 2004.
- [2] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone, *Classification and Regression Trees*, Belmont CA: Wadsworth, 1993.
- [3] J. Dvořák and P. Savický, “Softening Splits in Decision Trees Using Simulated Annealing”, *Adaptive and Natural Computing Algorithms*, LNCS vol. 4431/2007, pp. 721–729, 2007.
- [4] T. Fawcett, “An introduction to ROC analysis”, *Pattern Recognition Letters*, vol. 27, pp. 861—874, 2006.
- [5] J.A. Nelder and R. Mead, “A simplex algorithm for function minimization.”, *Computer Journal* vol. 7, pp. 308–313, 1965.
- [6] R Development Core Team (2008), “R: A language and environment for statistical computing”, R Foundation for Statistical Computing, Vienna, Austria. URL <http://www.r-project.org>.