Joanna Trzęsiok

ZASTOSOWANIE AGREGACYJNEJ METODY MART W DYSKRYMINACJI

Wprowadzenie

Przedmiotem analizy dyskryminacji jest zbiór obserwacji:

$$U = \{(\mathbf{x}_i, y_i) : i = 1, 2, \ldots, N\} \tag{1}$$

którego każdy element jest charakteryzowany przez wektor $p+1$ zmiennych:

$$(X_1, X_2, \ldots, X_p, Y)$$

Rozważamy zagadnienie dyskryminacji K klas, w którym zmienna zależna Y wyraża numer klasy, zatem:

$$y_i \in \{1, 2, \ldots, K\} \quad \text{dla} \quad i = 1, 2, \ldots, N \tag{2}$$

Celem analizy jest znalezienie funkcji F – reguły dyskryminującej, opisującej zależność zmiennej Y od zmiennych objaśniających:

$$Y = F(\mathbf{X}) + \varepsilon \tag{3}$$

gdzie $\mathbf{X} = (X_1, X_2, \ldots, X_p)$, natomiast ε jest składnikiem losowym.

W opracowaniu zostanie przedstawione nieklasyczne podejście do zagadnienia (3), jakim jest metoda MART wykorzystująca drzewa dyskryminacyjne.

Metodę MART (*multiple additive regression trees*) zaproponował Friedman w 1999 r. [2]. Jest to jedna z metod agregacyjnych (*boosting methods*), polegająca na tworzeniu modelu z wielu funkcji dyskryminujących lub funkcji regresji. Jest przykładem podejścia sekwencyjnego, w którym funkcje składowe, tworzone w każdym kroku, są systematycznie poprawiane. W efekcie prowadzi to do uzyskania modelu końcowego charakteryzującego się dużą dokładnością predykcji.

Metoda MART pozwala na wprowadzenie do modelu cech zarówno metrycznych, jak i niemetrycznych. Nie jest też wymagana znajomość rozkładów zmiennych objaśniających. Dopuszcza się także, aby rozkłady te charakteryzowały się wysoką skośnością. W metodzie MART nie ma również potrzeby standaryzacji zmiennych.

Omawiana metoda ma zastosowanie w regresji i dyskryminacji, jednak w opracowaniu zaprezentujemy jej wykorzystanie tylko w zadaniach dyskryminacji.

1. Algorytm *AdaBoost*

Jednym z pierwszych algorytmów dyskryminacyjnych wykorzystujących podejście wielomodelowe był *AdaBoost*, zaproponowany w 1997 r. przez Freunda i Schapire'a [1]. Oddaje on bardzo dobrze ideę metod agregacyjnych, dlatego zostanie zaprezentowany w tym opracowaniu.

AdaBoost ma zastosowanie do problemów dyskryminacji w przypadku dwóch klas. Oznaczmy przez Y numer klasy:

$$Y \in \{-1,1\}$$

Niech $\mathbf{X}$ będzie wektorem zmiennych objaśniających, zaś $f(\mathbf{X})$ – regułą dyskryminującą. Jakość dopasowania modelu będziemy mierzyć za pomocą błędu klasyfikacji:

$$err = \frac{1}{N}\sum_{i=1}^{N} I(y_i \neq f(\mathbf{x}_i)) \tag{4}$$

W algorytmie *AdaBoost* szukamy najpierw reguły dyskryminującej $f_1(\mathbf{X})$, wykorzystując zbiór uczący U. Następnie zbiór ten zostaje przekształcony – wszystkim obserwacjom zostają nadane odpowiednie wagi:

- obserwacje błędnie sklasyfikowane otrzymują wyższe wagi,
- obserwacje dobrze sklasyfikowane otrzymują niższe wagi.

Dla danych ważonych z przekształconego zbioru U ponownie szukamy reguły dyskryminującej $f_2(\mathbf{X})$. Po czym nadajemy wagi obserwacjom i powtarzamy procedurę aż do momentu, kiedy jakość modelu otrzymanego w kolejnym kroku nie różni się istotnie od jakości modelu z kroku poprzedniego. Z ciągu uzyskanych reguł $f_m(\mathbf{X})$ tworzymy zagregowany model.

Kolejne etapy algorytmu *AdaBoost* są następujące:

Algorytm 1

1. Przypisujemy wszystkim obserwacjom ze zbioru U wagi $w_i = \frac{1}{N}$ (dla $i = 1,...,N$).
2. Dla $m = 1,...,M$ wykonujemy kroki:

a) Dla danych ważonych przez $\{w_i : i = 1,...,N\}$ szukamy reguły dyskryminacyjnej $f_m(\mathbf{X})$.

b) Obliczamy błąd klasyfikacji:

$$err_m = \frac{\sum_{i=1}^{N} w_i \cdot I(y_i \neq f_m(\mathbf{X}))}{\sum_{i=1}^{N} w_i} \tag{5}$$

c) Obliczamy współczynniki:

$$\alpha_m = \log \frac{1 - err_m}{err_m} \tag{6}$$

d) Przypisujemy nowe wartości wagom:

$$w_i \leftarrow w_i \cdot \exp(\alpha_m \cdot I(y_i \neq f_m(\mathbf{X}))), \quad \text{dla} \quad i = 1,...,N \tag{7}$$

3. Końcowa reguła dyskryminująca ma postać:

$$F(\mathbf{X}) = sign\left[\sum_{m=1}^{M} \alpha_m \cdot f_m(\mathbf{X})\right] \tag{8}$$

Model końcowy (8) jest skonstruowany na podstawie kombinacji liniowej modeli składowych $f_m(\mathbf{X})$. Współczynniki α_m dane wzorem (6) wyrażają wpływ danej reguły dyskryminacyjnej $f_m(\mathbf{X})$ na postać funkcji $F(\mathbf{X})$ i zależą od błędu klasyfikacji dla danej reguły. Im większy błąd klasyfikacji err_m, tym mniejszy jest współczynnik α_m i tym samym składowa $f_m(\mathbf{X})$ ma mniejszy wpływ na postać końcową reguły dyskryminacyjnej.

2. Ogólna postać modelu zagregowanego

Model (8) otrzymany za pomocą algorytmu *AdaBoost* można uogólnić i zapisać w postaci addytywnej:

$$F(\mathbf{X}) = \sum_{m=0}^{M} \beta_m \cdot b(\mathbf{X}, \gamma_m) \tag{9}$$

gdzie funkcja składowa $b(\mathbf{X}, \gamma_m)$ jest funkcją dyskryminującą charakteryzowaną przez zbiór parametrów oznaczonych przez γ_m, zaś β_m to współczynnik rozwoju (*expension coefficient*) określający wagę składowej $b(\mathbf{X}, \gamma_m)$.

Dla modelu (9) ustalamy *a priori* postać funkcji b, natomiast estymatory parametrów modelu β_m i γ_m uzyskujemy przez minimalizację funkcji straty $L(y, F(\mathbf{X}))$ w zbiorze uczącym U:

$$\min_{\{\beta_m,\gamma_m\}_{m=1,\ldots,M}} \sum_{i=1}^{N} L\left(y_i, \sum_{m=0}^{M} \beta_m \cdot b(\mathbf{x}_i, \gamma_m)\right) \tag{10}$$

2.1. Minimalizacja funkcji straty – metoda największego spadku

Zagadnienie minimalizacji (10) jest złożone obliczeniowo, dlatego do jego rozwiązania stosuje się często strategię wspinaczki (*forward stagewise modeling*). Polega ona na wyborze w danym kroku rozwiązania optymalnego jedynie w sensie lokalnym. Jej zaletą jest prostota obliczeniowa, a dobrze określone kryterium może w rezultacie prowadzić do rozwiązania bliskiego optymalnemu w sensie globalnym.

W każdym kroku algorytmu wykorzystującego strategię wspinaczki szukamy estymatorów $\hat{\beta}_m$, $\hat{\gamma}_m$ dla pojedynczej funkcji składowej, posługując się funkcjami uzyskanymi w etapach poprzednich:

$$(\hat{\beta}_m, \hat{\gamma}_m) = \arg\min_{\{\beta,\gamma\}} \sum_{i=1}^{N} L\left(y_i, f_{m-1}(\mathbf{x}_i) + \beta \cdot b(\mathbf{x}_i, \gamma)\right) \tag{11}$$

Po czym przyjmujemy:

$$f_m(\mathbf{X}) = f_{m-1}(\mathbf{X}) + \hat{\beta}_m \cdot b(\mathbf{X}, \hat{\gamma}_m) \tag{12}$$

Rozwiązaniem zadania (11) są estymatory $\hat{\beta}_m$, $\hat{\gamma}_m$, które powodują największą redukcję wartości funkcji straty L. Koncepcja ta została powtórzona w metodzie największego spadku [2]. Tutaj, aby znaleźć minimum funkcji L, będziemy przesuwać się w kierunku przeciwnym do jej gradientu:

$$-\rho_m \cdot \mathbf{g}_m = -\rho_m \cdot [g_m(\mathbf{x}_i)]_{i=1,\ldots,N} \tag{13}$$

gdzie $\rho_m \in \mathbf{R}$ jest parametrem nazywanym „długością kroku", zaś $\mathbf{g}_m \in \mathbf{R}^N$ – gradientem z funkcji straty L o składowych:

$$g_m(\mathbf{x}_i) = \left[\frac{\partial L(y_i, F(\mathbf{x}_i))}{\partial F(\mathbf{x}_i)}\right]_{F(\mathbf{x}_i)=f_{m-1}(\mathbf{x}_i)} \tag{14}$$

Algorytm metody największego spadku możemy zapisać w następujących krokach:

Algorytm 2

1. Przyjmujemy jako model początkowy:

$$f_0(\mathbf{x}) = \arg\min_{\rho} \sum_{i=1}^{N} L(y_i, \rho) \tag{15}$$

2. Dla $m = 1,\ldots,M$ wykonujemy kroki:

   a) Obliczamy składowe gradientu (pseudoreszty):

$$\tilde{y}_i = g_m(\mathbf{x}_i) = \left[\frac{\partial L(y_i, F(\mathbf{x}_i))}{\partial F(\mathbf{x}_i)}\right]_{F(\mathbf{x}_i)=f_{m-1}(\mathbf{x}_i)} \quad \text{dla } i = 1,\ldots,N \tag{16}$$

   b) Wyznaczamy estymatory parametrów modelu składowego:

$$\hat{\gamma}_m = \arg\min_{\{\beta,\gamma\}} \sum_{i=1}^{N} \left(\tilde{y}_i - \beta \cdot b(\mathbf{x}_i, \gamma)\right)^2 \tag{17}$$

   c) Wyznaczamy parametr „długości kroku":

$$\rho_m = \arg\min_{\rho} \sum_{i=1}^{N} L\left(y_i, f_{m-1}(\mathbf{x}_i) + \rho \cdot b(\mathbf{x}_i, \hat{\gamma}_m)\right) \tag{18}$$

   d) Przyjmujemy:

$$f_m(\mathbf{X}) = f_{m-1}(\mathbf{X}) + \rho_m \cdot b(\mathbf{X}, \hat{\gamma}_m) \tag{19}$$

3. Model końcowy ma postać:

$$F(\mathbf{X}) = f_M(\mathbf{X}) \tag{20}$$

W pierwszym kroku algorytmu 2 został wyznaczony model początkowy, który w tym przypadku jest funkcją stałą. Następnie w każdej iteracji obliczamy pseudoreszty (16), które powodują największy spadek wartości funkcji straty L. Na ich podstawie wyznaczamy estymatory parametrów modelu składowego (17), a następnie parametr „długości kroku" (18). Posługując się otrzymanymi wielkościami (17), (18) i modelem składowym otrzymanym w poprzednim etapie, tworzymy funkcję f_m. Model końcowy jest modelem uzyskanym w ostatnim kroku algorytmu.

Procedurę wyznaczania parametrów modelu (9), przez minimalizację funkcji straty z wykorzystaniem metody największego spadku, nazywamy gradientową metodą agregacyjną (*gradient boosting method*).

3. Metoda MART w dyskryminacji

Rozpatrywana w tym opracowaniu metoda MART będzie dotyczyć przypadku dyskryminacji K klas. Załóżmy, że:

$$y_k = \begin{cases} 1, & \text{gdy } \mathbf{x}_k \text{ znajduje się w klasie } k \\ 0, & \text{w przeciwnym wypadku} \end{cases} \tag{21}$$

Oznaczmy przez $p_k(\mathbf{x})$ prawdopodobieństwo tego, że obiekt $\mathbf{x}$ należy do klasy k:

$$p_k(\mathbf{x}) = \Pr(y_k = 1 \mid \mathbf{x}) \tag{22}$$

Przyjmujemy regułę dyskryminującą $F_k(\mathbf{X})$ dla klasy o numerze k, wykorzystującą transformację logistyczną, w postaci:

$$F_k(\mathbf{X}) = \ln p_k(\mathbf{X}) - \frac{1}{K}\sum_{l=1}^{K} \ln p_l(\mathbf{X}) \tag{23}$$

Przekształcając formułę (23), określającą funkcję $F_k(\mathbf{X})$, otrzymujemy równanie wyrażające prawdopodobieństwo $p_k(\mathbf{X})$:

$$p_k(\mathbf{X}) = \frac{\exp(F_k(\mathbf{X}))}{\sum_{l=1}^{K} \exp(F_l(\mathbf{X}))} \tag{24}$$

Metoda MART wykorzystywana w dyskryminacji jest jedną z gradientowych metod agregacyjnych. Model otrzymany za jej pomocą możemy zapisać w postaci addytywnej danej równaniem (9). Jako funkcje b w tym modelu będziemy przyjmować drzewa dyskryminacyjne. Natomiast estymatory parametrów funkcji składowych otrzymamy minimalizując funkcję straty. Posłużymy się tutaj opisaną wcześniej metodą największego spadku.

3.1. Funkcje składowe modelu

Tak jak już zostało wspomniane, w omawianej metodzie MART w dyskryminacji jako funkcje składowe b będziemy przyjmować drzewa dyskryminacyjne.

Metoda drzew dyskryminacyjnych polega na podziale przestrzeni zmiennych objaśniających na rozłączne obszary (hiperkostki) R_j (dla $j = 1,...,J$) i przypisaniu wszystkim obserwacjom z danego obszaru odpowiedniej wartości γ_j:

$$\mathbf{x} \in R_j \Rightarrow f(\mathbf{x}) = \gamma_j \quad \text{dla} \quad j = 1,...,J \tag{25}$$

W zagadnieniu dyskryminacji γ_j jest numerem klasy otrzymanym za pomocą reguły majoryzacji.

Model otrzymany za pomocą drzew dyskryminacyjnych można zapisać w postaci addytywnej:

$$T(\mathbf{X}, \Theta) = \sum_{j=1}^{J} \gamma_j \cdot I(\mathbf{X} \in R_j) \tag{26}$$

z parametrami $\Theta = \{R_j, \gamma_j\}_{j=1,...,J}$.

Natomiast model zagregowany złożony z drzew $T(\mathbf{x}, \Theta_m)$ ma postać:

$$F(\mathbf{X}) = \sum_{m=0}^{M} T(\mathbf{X}, \Theta_m) = \sum_{m=0}^{M} \sum_{j=1}^{J_m} \gamma_{jm} \cdot I(\mathbf{X} \in R_{jm}) \tag{27}$$

gdzie $\Theta_m = \{R_{jm}, \gamma_{jm}\}_{m=0,...,M,\ j=1,...,J_m}$.

Estymatory parametrów Θ_m w modelu zagregowanym (27) otrzymujemy przez minimalizację funkcji straty:

$$\hat{\Theta}_m = \underset{\{\Theta_m\}_{m=0,...,M}}{\arg\min} \sum_{i=1}^{N} L\left(y_i, \sum_{m=0}^{M} T(\mathbf{x}_i, \Theta_m) \right) \tag{28}$$

3.2. Postać funkcji straty

W metodzie MART do wyznaczenia parametrów modelu zagregowanego (27), zapisanego w postaci drzew dyskryminacyjnych, posłużymy się algorytmem 2 – metodą największego spadku.

Funkcja straty L wykorzystywana w zagadnieniu dyskryminacji K klas ma postać:

$$L((y_k, F_k(\mathbf{X}))_{k=1,...,K}) = -\sum_{k=1}^{K} y_k \ln p_k(\mathbf{X}) \tag{29}$$

gdzie y_k oraz prawdopodobieństwo $p_k(\mathbf{X})$ są opisane wzorami (21), (22).

Minimalizując funkcję straty za pomocą strategii wspinaczki, wyznaczamy składowe gradientu funkcji L:

$$\left[\frac{\partial L((y_{il}, F_l(\mathbf{x}_i))_{l=1,\ldots,K})}{\partial F_k(\mathbf{x}_i)}\right]_{(F_l(\mathbf{x})=f_{l,m-1}(\mathbf{x}))_{l=1,\ldots,K}} \tag{30}$$

Podstawiając do równania (29) prawdopodobieństwo $p_k(\mathbf{X})$ dane wzorem (24) i obliczając gradient funkcji L, otrzymujemy następujące pseudoreszty:

$$\tilde{y}_{ik} = -\left[\frac{\partial L((y_{il}, F_l(\mathbf{x}_i))_{l=1,\ldots,K})}{\partial F_k(\mathbf{x}_i)}\right]_{(F_l(\mathbf{x})=f_{l,m-1}(\mathbf{x}))_{l=1,\ldots,K}} = y_{ik} - p_{k,m-1}(\mathbf{x}_i) \tag{31}$$

gdzie $p_{k,m-1}(\mathbf{X})$ jest pochodną funkcji $F_{k,m-1}(\mathbf{X})$.

3.3. Algorytm MART

Algorytm MART jest pewnym szczególnym przypadkiem algorytmu 2, opartym na drzewach dyskryminacyjnych, wykorzystującym funkcje straty w postaci (29). Algorytm ten można zapisać w następujących krokach [2]:

1. Przyjmujemy początkowe funkcje dyskryminujące:

$$F_{k0}(\mathbf{X}) = 0, \quad \text{dla} \quad k = 1,\ldots,K \tag{32}$$

2. Dla $m = 1,\ldots,M$ wykonujemy kroki:
   a) Obliczamy prawdopodobieństwo:

$$p_k(\mathbf{X}) = \frac{\exp(F_k(\mathbf{X}))}{\sum_{l=1}^{K}\exp(F_l(\mathbf{X}))}, \quad \text{dla} \quad k = 1,\ldots,K \tag{33}$$

   b) Dla $k = 1,\ldots,K$
      - obliczamy pseudoreszty:

$$\tilde{y}_{ik} = y_{ik} - p_k(\mathbf{x}_i), \quad \text{dla} \quad i = 1,\ldots,N \tag{34}$$

      - szukamy obszarów R_{jkm} (dla $j = 1,\ldots,J$), posługując się zbiorem:

$$\{(\mathbf{x}_i, \tilde{y}_{ik}) : i = 1,\ldots,N\} \tag{35}$$

      - wyznaczamy parametry modelu:

$$\gamma_{jkm} = \frac{K-1}{K} \cdot \frac{\sum_{\mathbf{x}_i \in R_{jkm}} \tilde{y}_{ik}}{\sum_{\mathbf{x}_i \in R_{jkm}} |\tilde{y}_{ik}|(1-|\tilde{y}_{ik}|)}, \quad \text{dla} \quad j = 1,\ldots,J \tag{36}$$

– przyjmujemy:

$$F_{km}(\mathbf{X}) = F_{k,m-1}(\mathbf{X}) + \sum_{j=1}^{J} \gamma_{jkm} \cdot I(\mathbf{X} \in R_{jkm}) \tag{37}$$

3. Końcowe reguły dyskryminujące mają postać:

$$F_k(\mathbf{X}) = F_{kM}(\mathbf{X}), \quad \text{dla} \quad k = 1,\ldots,K \tag{38}$$

W pierwszym kroku tego algorytmu ustalamy początkowe reguły dyskryminujące (32), a następnie wyznaczamy prawdopodobieństwa $p_k(\mathbf{X})$ zgodnie ze wzorem (24). W kolejnym etapie obliczamy pseudoreszty $\tilde{y}_{ik}$, wyznaczające największy spadek funkcji straty, do których dopasowujemy drzewo dyskryminacyjne, po czym poprawiamy modele składowe i powtarzamy procedurę.

4. Przykład zastosowania metody MART w dyskryminacji

Dla zilustrowania metody MART przeprowadzono obliczenia na dwóch zbiorach danych: *Iris* i *Vehicle*, standardowo wykorzystywanych do badania własności nieklasycznych metod dyskryminacji. Obliczenia zostały wykonane za pomocą programu *TreeNet*.

4.1. Zbiór danych *Iris*

Dane w zbiorze *Iris* zostały zebrane w 1936 r. przez Fishera. Zbiór ten zawiera 150 obserwacji dotyczących kwiatów irysa. Są one charakteryzowane przez cztery zmienne objaśniające:

- długość działki kielicha (*sepallen*),
- szerokość działki kielicha (*sepalwid*),
- długość płatka (*petalwid*),
- szerokość płatka (*petallen*).

W badanym zbiorze znajdują się trzy gatunki kwiatów: Setosa, Versicolor, Virginica.

W prezentowanym przykładzie cały zbiór *Iris* został wykorzystany jako zbiór uczący. W celu wygenerowania modelu w algorytmie MART wykonano 1000 iteracji. Wyniki dyskryminacji przeprowadzonej na zbiorze *Iris* zaprezentowano w tab. 1. Metoda MART pozwoliła na całkowite wyodrębnienie klasy

Setosa, natomiast kilka kwiatów z gatunku Virginica zostało zaklasyfikowanych do Versicolor i odwrotnie. Błąd klasyfikacji obliczony na zbiorze uczącym wynosi:

$$err = 0{,}027$$

Tabela 1

Wynik klasyfikacji na zbiorze uczącym dla zbioru *Iris*

Klasa	Liczba elementów	Procent elementów właściwie zaklasyfikowanych	1 N = 50	2 N = 48	3 N = 52
Setosa	50	100,000	50	0	0
Versicolor	50	94,000	0	47	3
Virginica	50	98,000	0	1	49

Metoda MART pozwala na stworzenie rankingu istotności wpływu zmiennych objaśniających na numer klasy (zob. tab. 2). W naszym przykładzie największą moc dyskryminującą ma zmienna *petalwid*, czyli długość płatka.

Tabela 2

Ranking istotności wpływu zmiennych objaśniających na numer klasy

Zmienna	Wynik	
Petalwid	100,00	
Petallen	49,42	
Sepalwid	7,73	
Sepallen	5,17	

Program *TreeNet* pokazuje również, na którą klasę dana zmienna ma największy wpływ. Odpowiednie wyniki przedstawiają tab. 3-6.

Tabela 3

Wpływ zmiennej objaśniającej *petalwid* na numer klasy

Zmienna	Wynik	
Setosa	96,71	
Versicolor	100,00	
Virginica	73,14	

Tabela 4

Wpływ zmiennej objaśniającej *petallen* na numer klasy

Zmienna	Wynik	
Setosa	2,48	
Versicolor	100,00	
Virginica	99,14	

Tabela 5

Wpływ zmiennej objaśniającej *sepalwid* na numer klasy

Zmienna	Wynik	
Setosa	0,27	
Versicolor	99,38	
Virginica	100,00	

Tabela 6

Wpływ zmiennej objaśniającej *sepallen* na numer klasy

Zmienna	Wynik	
Setosa	12,23	
Versicolor	100,00	
Virginica	94,71	

4.2. Zbiór danych *Vehicle*

Zbiór danych *Vehicle* zawiera 864 obserwacje charakteryzowane przez 19 metrycznych zmiennych objaśniających. Obiekty są przydzielone do czterech klas.

W programie *TreeNet* badany zbiór został losowo podzielony na zbiór uczący i testowy (stanowiący 20% całego zbioru). W celu wygenerowania modelu w algorytmie MART wykonano 1000 iteracji. Otrzymane wyniki dla zbioru uczącego (tab. 7) i testowego (tab. 8) przedstawiono poniżej.

Tabela 7

Wynik klasyfikacji na zbiorze uczącym dla zbioru *Vehicle*

Klasa	Liczba elementów	Procent elementów właściwie zaklasyfikowanych	1 N = 163	2 N = 179	3 N = 175	4 N = 152
1	164	98,780	162	2	0	0
2	178	99,438	1	177	0	0
3	175	100,000	0	0	175	0
4	152	100,000	0	0	0	152

Tabela 8

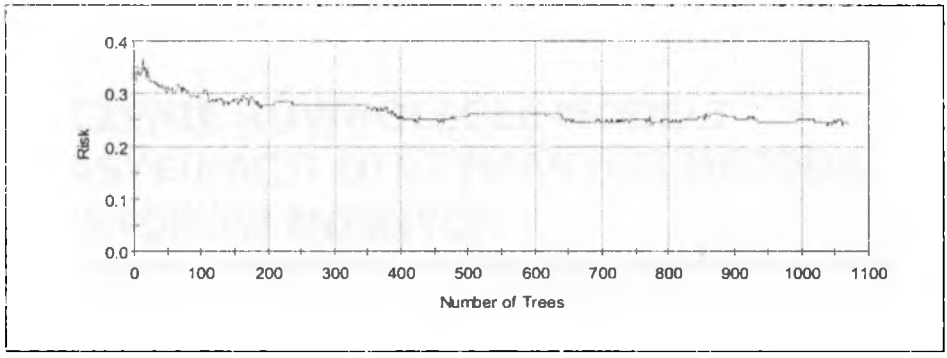
Wynik klasyfikacji na zbiorze testowym dla zbioru *Vehicle*

Klasa	Liczba elementów	Procent elementów właściwie zaklasyfikowanych	1 N = 48	2 N = 36	3 N = 42	4 N = 51
1	48	58,333	28	15	0	5
2	39	48,718	19	19	0	1
3	43	97,674	0	1	42	0
4	47	95,745	1	1	0	45

Wyniki dyskryminacji wykonanej na zbiorze testowym pokazują, że największy problem stanowi właściwe oddzielenie klasy pierwszej od drugiej. Błąd klasyfikacji obliczony na zbiorze testowym wynosi:

$$err = 0{,}243$$

Rysunek 1 przedstawia błąd klasyfikacji obliczony dla zbioru testowego w kolejnych iteracjach metody MART. Jak widzimy, począwszy od około 200 iteracji błąd klasyfikacji ustabilizował się na poziomie około 0,25, tym samym zwiększanie liczby iteracji nie jest celowe, gdyż nie prowadzi do polepszenia jakości modelu.



Rys. 1. Błąd klasyfikacji obliczony na zbiorze testowym w kolejnych etapach algorytmu MART

Literatura

1. Freund Y., Schapire R.: *A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting*. „Journal of Computer and System Sciences" 1997, 55, s. 119-139.
2. Friedman J.H.: *Greedy Function Approximation: A Gradient Boosting Machine*. Technical report, Dept. of Statistics, Stanford University, 2001.
3. Friedman J.H., Hastie T., Tibshirani R.: *Additive Logistic Regression: A Statistical View of Boosting*. Technical report, Dept. of Statistics, Stanford University, 1999.
4. Gatnar E.: *Nieparametryczna metoda dyskryminacji i regresji*. Wydawnictwo Naukowe PWN, Warszawa 2000.
5. Hastie T., Tibshirani R., Friedman J.H.: *The Elements of Statistical Learning*. Springer-Verlag, New York 2001.

USAGE OF THE BOOSTING METHOD MART IN DISCRIMINATION

Summary

Multiple additive regression trees MART belong to the group of boosting methods for regression and classification. This approach was introduced by J.H. Friedman (1999). Besides the accuracy, its primary goal is robustness. It tends to be resistant against the outliers, missing values, and the inclusion of the potentially large numbers of irrelevant predictor variables that have little or no effect on the response.

In this paper the MART algorithm the for classification and its applications has been presented.