# Hierarchiczna reprezentacja wiedzy dla automatycznego podejmowania decyzji

**Janusz Będkowski, Andrzej Masłowski**
Instytut Automatyki i Robotyki Politechniki Warszawskiej

W pracy przedstawiono ideę hierarchicznej reprezentacji wiedzy dla automatycznego podejmowania decyzji. Hierarchiczna reprezentacja wiedzy została zaproponowana do modelowania predykcji. Pokazano efektywne podejmowanie decyzji na przykładzie klasyfikacji zbioru danych, który nie jest separowany liniowo. Warto podkreślić, że nie założono wiedzy a priori o zbiorze danych oraz relacji między elementami tego zbioru oraz że proponowany algorytm automatycznie odkrywa optymalne granice decyzyjne między nimi. Przedstawiono algorytm konstrukcji hierarchicznej reprezentacji wiedzy, który wprowadza ocenę jakościową powstałej struktury na poszczególnych poziomach decyzyjnych. Przeprowadzony eksperyment numeryczny pokazuje zalety proponowanego algorytmu, który może być wykorzystany do zadań klasyfikacji, gdzie występuje problem doboru algorytmu klasyfikacji.

**Słowa kluczowe:** sieć neuronowa, maszyna wektorów nośnych, drzewo decyzyjne, hierarchiczna reprezentacja wiedzy

Hierarchiczna reprezentacja wiedzy jest bazą wielu systemów podejmowania decyzji [8, 9, 10]. Jednym z problemów spotykanych w modelowaniu złożonych systemów podejmowania decyzji jest znaczna liczba reguł decyzyjnych oraz złożoność relacji logicznych między nimi. Z punktu widzenia inteligencji obliczeniowej zaproponowano już wiele aplikacji hierarchicznej reprezentacji wiedzy, m.in. sieci Hierarchical Fuzzy Petri Net [11, 12]. Taki system podejmowania decyzji dekomponuje zbiór elementów na mniejsze podzbiory zorganizowane w poziomy, między którymi występują zdefiniowane zależności funkcyjne. W pracy [4] przedstawiono możliwość wykorzystania hierarchicznej reprezentacji wiedzy w realizacji zadania aproksymacji funkcji. Przedstawiono praktyczny przykład systemu sieci neuronowych, który umożliwia budowę hierarchicznej reprezentacji wiedzy. W pracy [5] przedstawiono framework eksploracji wiedzy także wykorzystujący koncepcję hierarchicznej reprezentacji wiedzy. Kolejne prace [6, 7] przedstawiają aplikacje działające on-line wspierające podejmowanie decyzji oraz nadzorujące decyzje człowieka. Nadzorowanie decyzji człowieka polega w tym przypadku na wspomaganiu operatora podczas wykonywania rutynowych czynności za pomocą adaptacyjnego systemu podejmowania decyzji.

W pracy przedstawiono nowy algorytm klasyfikujący dane, który w odróżnieniu od opracowanych rozwiązań może integrować różne klasyfikatory z rodziny sieci neuronowych czy maszyn wektorów nośnych w celu wybrania optymalnej granicy decyzyjnej na danym poziomie hierarchicznej struktury reprezentacji wiedzy (drzewa decyzyjnego). Proponowane podejście może mieć zastosowanie w przypadkach, gdzie nie jest dostępna żadna przesłanka co do wyboru klasyfikatora. W wyniku procedury automatycznej oceny klasyfikatorów na każdym poziomie decyzyjnym wybierana jest optymalna granica decyzyjna na bazie przyjętych kryteriów oceny.

W pracy przedstawiono algorytm hierarchicznej reprezentacji wiedzy w postaci drzewa decyzyjnego, w którym kolejne poziomy są skojarzone za pomocą granic decyzyjnych wyznaczonych przez algorytmy SVM (*Support Vector Machine*) oraz MPL (*Multi Layer Perceptron*). Przedstawiono algorytm wyboru testu, który automatycznie wybiera kandydata realizującego granice decyzyjne spośród określonego zbioru dostępnych testów. Przedstawiono numeryczny eksperyment pokazujący ideę proponowanego algorytmu.

W artykule przedstawiono klasyczne algorytmy klasyfikujące dane z rodziny sieci neuronowych i maszyn wektorów nośnych oraz algorytm drzewa decyzyjnego z uwzględnieniem procedury obliczenia parametru entropii. Pokazano też modyfikację drzewa decyzyjnego poprzez rozbudowę zbioru dostępnych testów o klasyfikatory SVM oraz MLP. Zademonstrowano klasyfikację zbioru danych należących do czterech różnych klas. W podsumowaniu skomentowano otrzymany wynik oraz wady i zalety proponowanego algorytmu klasyfikacji danych.

## Klasyfikator – sieć neuronowa MLP

MLP (*Multi Layer Perceptron*) jest klasycznym klasyfikatorem z rodziny sieci neuronowych. Realizowana przez neuron funkcja separująca dane na dwa podzbiory jest określona wzorem:

$$y(x) = \sum_{j=0}^{n} w_j x_j = 0 \tag{1}$$

Po przekształceniu funkcja wygląda następująco:

$$y(x) = \sum_{j=1}^{n} w_j x_j + w_0 = \mathbf{w}^T \mathbf{x} + w_o = 0 \tag{2}$$

W konsekwencji równanie $y(x) = 0$ określa granicę decyzyjną separującą dane należące do dwóch różnych klas. Dla przykładu dla $n = 2$ równanie (2) przyjmuje postać:

$$w_1 x_1 + w_2 x_2 + w_0 = 0 \tag{3}$$

Granica decyzyjna wyznaczona przez (3) jest prostą o równaniu:

$$x_2 = -\frac{w_1}{w_2} x_1 - \frac{w_0}{w_2} \tag{4}$$

Warto podkreślić, że każdy dodatkowy neuron w warstwie ukrytej wprowadza dodatkową granicę decyzyjną. Po złożeniu pojedynczych granic decyzyjnych powstaje wynikowa granica decyzyjna.

## Maszyna Wektorów Nośnych

Maszyna wektorów nośnych [13, 14, 15] SVM (*Support Vector Machine*) to stosunkowo nowy, uczący się system wykonujący zadanie klasyfikacji obiektów oraz aproksymacji funkcji na podstawie danych pomiarowych. Maszyny wektorów nośnych należą do rodziny tzw. maszyn z jądrem liniowym bądź nieliniowym (*kernel machines*). Maszyna wektorów nośnych rozwiązuje zadania klasyfikacji i regresji metodami optymalizacji z ograniczeniami. W zadaniu klasyfikacji poszukujemy optymalnej hiperpłaszczyzny rozdzielającej dane należące do różnych klas. Ideę maszyny wektorów nośnych można najprościej przedstawić dla przypadku dwóch klas liniowo separowalnych. Dany zbiór S punktów $\mathbf{x_i} \in R^n$, $i = 1, 2, \ldots, N$ jest *liniowo separowalny*, jeżeli istnieją wektor $\mathbf{w} \in R^n$ oraz liczba $b$ takie, że:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \quad i=1,2,...,N \tag{5}$$

przy czym $y_i \in [-1, 1]$ jest etykietą klasy. Hiperpłaszczyzna rozdzielająca przestrzeń wejściową na dwie podprzestrzenie jest określona zależnością:

$$(\mathbf{w} \cdot \mathbf{x}) + b = 0 \tag{6}$$

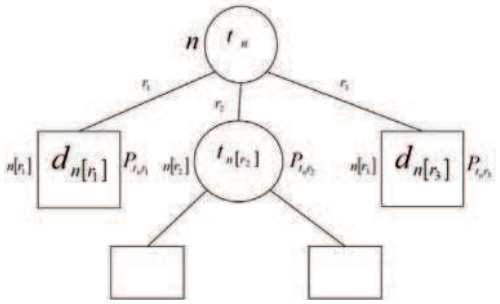
gdzie **w** jest wektorem kierunkowym hiperpłaszczyzny, a liczba $b$ jest równa odległości hiperpłaszczyzny od początku układu współrzędnych. Para ($\mathbf{w}$, $b$) określa jednoznacznie równanie optymalnej hiperpłaszczyzny rozdzielającej. Klasyfikator liniowy jest określony zależnością:

$$f(\mathbf{x}) = (\mathbf{w} \cdot \mathbf{x}) + b = \sum_{i=1}^{n} w_i x_i + b \tag{7}$$

Jeżeli $f(\mathbf{x}) \geq 0$, to wejście **x** należy do klasy dodatniej, w przeciwnym przypadku – do klasy ujemnej.

## Drzewo Decyzyjne

Przez drzewo decyzyjne opisuje się strukturę [16] złożoną z węzłów, od których odchodzą gałęzie. Formalnie drzewo decyzyjne określa się jako spójny, skierowany graf acykliczny, przy czym krawędzie nazywamy gałęziami, wierzchołki, których wychodzi co najmniej jedna krawędź, są nazywane węzłami, a pozostałe wierzchołki liśćmi. Przyjmuje się, że każdy wierzchołek ma co najmniej jedną krawędź wchodzącą, a liczba krawędzi wychodzących jest 0 bądź większa od 1. Węzły, gałęzie i liście mają dodatkową interpretację. Pierwsze odpowiadają testom przeprowadzanym na wartościach atrybutów przykładów, drugie odpowiadają wynikom tych testów, ostatnie odpowiadają etykietom kategorii. Węzeł lub liść, który nie ma żadnych węzłów macierzystych, jest korzeniem drzewa. Drzewo decyzyjne przestawiono na rys. 1.




**Rys. 1.** Drzewo decyzyjne [16]
**Fig. 1.** Decision tree [16]

Dla dowolnego przykładu $x \in X$ istnieje dokładnie jeden związany z nim liść drzewa T, który oznaczony jest przez $l_{T,x}$ oraz zbiór odpowiadających mu na różnych poziomach węzłów, oznaczony przez $N_{T,x}$. Dla dowolnego węzła $n \in N_T$ drzewa T i zbioru przykładów $P \subseteq X$ jego podzbiór związany z $n$ oznaczony jest przez $P_{T,n}$ oraz dla dowolnego liścia $l \in L_T$ zbiór odpowiadających mu przykładów z $P$ przez $P_{T,l}$. Dla dowolnego zbioru przykładów $P$ i testu $t$, ze względu na różne wyniki testu $t$ wynikowy podzbiór przybiera postać:

$$P_{tr} = \left\{ x \in P \middle| t(x) = r \right\}, r \in R_t$$
$$P_{tr}^d = P^d \cap P_{tr}, d \in C, r \in R_t \tag{8}$$

Dla dowolnego węzła $n$ drzewa decyzyjnego przez $t_n$ oznaczony jest związany z tym węzłem test. Dla każdego z wyników testu $r \in R_{t_n}$ przez $n[r]$ oznaczony jest węzeł lub liść potomny, do którego prowadzi z węzła $n$ gałąź odpowiadająca wynikowi $r$. Dla dowolnego liścia $l$ przez $d_l$ oznaczona jest związana z tym liściem etykieta kategorii pojęcia docelowego. Drzewo decyzyjne realizuje określenie reguły jako wyrażenia postaci:

JEŚLI *warunki* TO *kategoria*,

które wyznacza kategorię przykładów, których wartości atrybutów spełniają podane warunki, na ogół stanowiące koniunkcję pewnych warunków elementarnych. Drzewo decyzyjne stanowi reprezentację pewnej hipotezy, ponadto jest także reprezentacją pewnego zbioru reguł. Drzewo decyzyjne może być skonstruowane za pomocą rekurencyjnej procedury zstępującego konstruowania drzewa. O utworzeniu liścia lub węzła decyduje kryterium zatrzymania procedury. Jeśli jest ono spełnione, to jest tworzony liść i na podstawie etykietowanych przykładów ze zbioru $P$ jest ustalana jego etykieta $d_1$, na czym proces budowy drzewa się kończy.

W przeciwnym przypadku jest tworzony węzeł $n$ – dokonuje się wyboru testu $t_n$ ze zbioru S, a następnie dla każdego z jego możliwych wyników odpowiednia gałąź węzła prowadzi do poddrzewa, które jest budowane przez rekurencyjne wywołanie tej samej procedury. W wywołaniu rekurencyjnym tworzącym poddrzewo, do którego ma prowadzić gałąź odpowiadająca wynikowi $r$ testu $t_n$, przekazywany jako argument zbiór przykładów zawierający tylko te przykłady z $P$, dla których test $t_n$ daje wynik $r$. W konsekwencji zbiór pozostających testów S jest pomniejszany o test $t_n$.

## Entropia

W teorii informacji entropia jest miarą zróżnicowania, np. zbioru danych. W tym ujęciu entropia jest dana wzorem:

$$E_{tr}(P) = \sum_{d \in C} -\frac{\left|P_{tr}^d\right|}{\left|P_{tr}\right|} \log \frac{\left|P_{tr}^d\right|}{\left|P_{tr}\right|} \tag{9}$$

gdzie: $P$ – zbiór danych, $d$ – etykieta kategorii, $C$ – zbiór etykiet, $t$ – test, $r$ – wynik testu $t$ zbioru danych $P$, $|P_{tr}^d|$ – liczba przykładów kategorii $d$ dla testu $t$, który daje wynik $r$, $|P_{tr}|$ – liczba przykładów dla testu $t$ dających wynik $r$.

Entropia zbioru dla zbioru $P$ jest dana wzorem:

$$E_t(P) = \sum_{r \in R_t} \frac{|P_{tr}|}{|P|} E_{tr}(P) \qquad (10)$$

gdzie: $R_i$ – zbiór możliwych wyników testu $t$, $r$ – wynik testu $t$, $E_{tr}$ – entropia zbioru $P$ dla test $t$ dającego wynik $r$, $|P_{tr}|$ – liczba przykładów dla testu $t$ dających wynik $r$, $|P|$ – liczba przykładów w zbiorze $P$. Dla każdego poziomu (węzła drzewa decyzyjnego) wartość entropii dla danego testu determinuje jego optymalny wybór. Minimalna wartość entropii gwarantuje, że węzeł drzewa zostanie podzielony na podzbiory, w których występują przykłady tych samych kategorii.

## Modyfikacja drzewa decyzyjnego

Rdzeniem algorytmu drzewa decyzyjnego jest test, na bazie którego podejmowana jest decyzja o podziale danego węzła na podzbiory przykładów, w których występują podobne kategorie. Modyfikacja algorytmu budowy drzewa decyzyjnego polega na wprowadzeniu nowych testów opartych na klasyfikatorach MLP oraz SVM.

Opracowano sposób oceny nowego testu, SVM oceniany jest pod względem liczby wektorów nośnych (im mniej wektorów nośnych tym lepszy model), wartości entropii (mniejsza wartość entropii gwarantuje podział na podzbiory danych tych samych kategorii) oraz skuteczność klasyfikacji (większa skuteczność oznacza, że model jest bardziej odpowiedni). Dąży się do wyboru możliwie najmniej złożonego modelu przy jednoczesnej minimalizacji entropii i maksymalizacji skuteczności.

Dla klasyfikatora typu MLP ocena jest dokonywana na podstawie liczby neuronów w warstwie ukrytej (mniejsza liczba oznacza, że model jest mniej złożony), wartości entropii oraz skuteczności klasyfikacji. Do procedury wyboru testu dodano także ograniczenie gwarantujące, że nie wystąpi nadmierne dopasowanie, tj. w przypadku SVM liczba wektorów nośnych < TOL × liczba przykładów, gdzie TOL < 0,1 gwarantuje sukces w większości przypadków. W przypadku klasyfikatora MLP liczba neuronów w warstwie ukrytej powinna być jak najmniejsza. Aby porównać najlepszego kandydata spośród SVM z najlepszym kandydatem spośród MLP, wykonuje się ocenę pod względem złożoności modelu. Wybierany jest model podziału przestrzeni o mniejszej złożoności obliczeniowej potrzebnej do wygenerowania decyzji. To kryterium wyboru powoduje, że drzewo decyzyjne jest zoptymalizowane pod względem czasu potrzebnego na wygenerowanie decyzji.

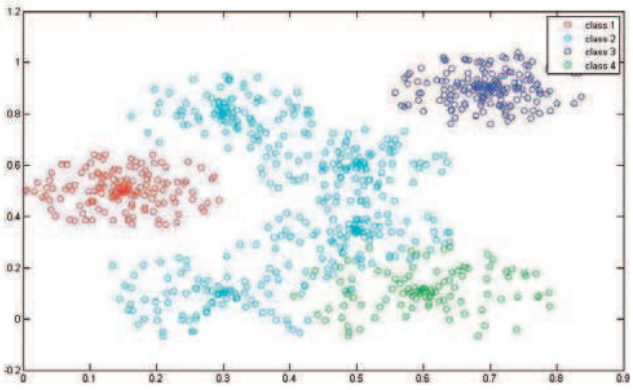
### Przykład

W przykładzie przedstawiono problem klasyfikacji zbioru punktów należących do czterech różnych klas (rys. 2) należących do płaszczyzny $X = \Re^2$. Punkty mają dwa atrybuty – $a_1$: $X \mapsto R$ oraz $a_2 : X \mapsto R$.

Na pierwszym poziomie drzewa decyzyjnego przyjęto podział na dwa podzbiory. Na rys. 3 pokazano siedem możliwości takiego podziału.

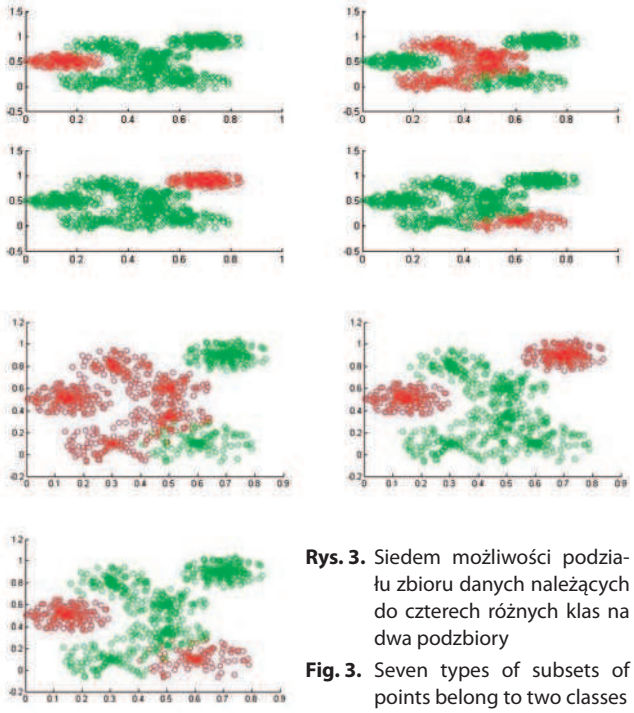
Korespondujące do rys. 3 drzewo decyzyjne przyjmuje postać przedstawioną na rys. 4.

*Root* – korzeń – reprezentuje główny element drzewa decyzyjnego. Z korzenia wychodzą dwie gałęzie. Wynika z tego, że



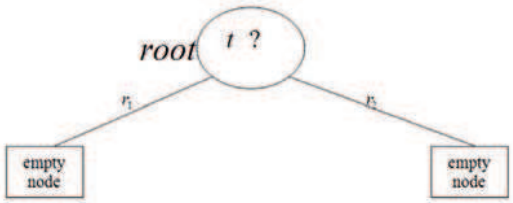

**Rys. 2.** Zbiory punktów należących do czterech klas

**Fig. 2.** Data set of points from four different classes



**Rys. 3.** Siedem możliwości podziału zbioru danych należących do czterech różnych klas na dwa podzbiory

**Fig. 3.** Seven types of subsets of points belong to two classes




**Rys. 4.** Pierwszy poziom drzewa decyzyjnego

**Fig. 4.** First level of the decision tree

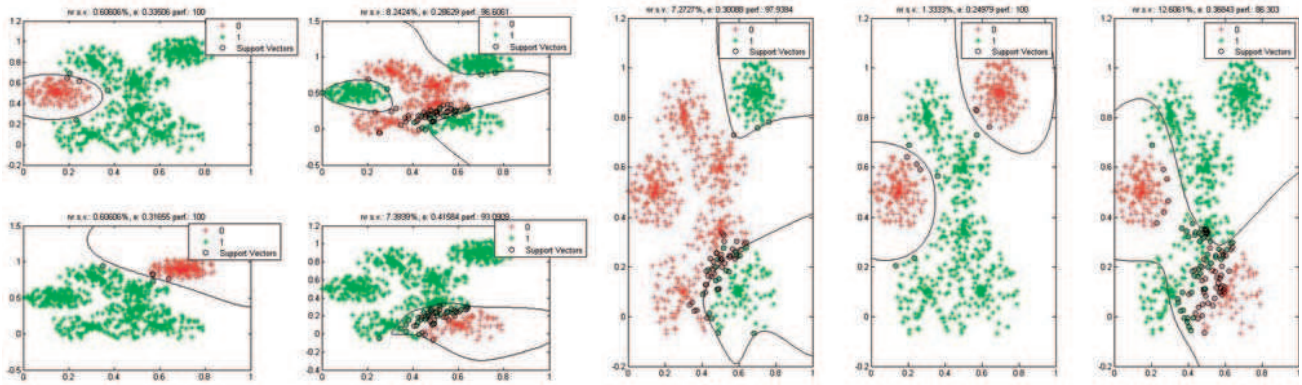
zbiór wyników $r$ składa się z dwóch elementów $\{r_1, r_2\}$. Należy wybrać optymalny test $t$ dzielący zbiór przykładów na dwa podzbiory. Aby wybrać odpowiedni test, wzięto pod uwagę siedem możliwości podziału przestrzeni danych. Dla każdej kombinacji obliczono model SVM korzystając z algorytmu SMO przy następujących parametrach:

KKTViolationLevel = 0,05 (parametr dla założenia, że 5 % danych może nie spełniać warunków KKT Karush-Kuhn-Tucker)

TolKKT = 0,1 (tolerancja brana pod uwagę przy sprawdzaniu warunków KKT)

Funkcja jądra – RBF sigma = 1.

Na rys. 5 przedstawiono siedem modeli SVM.



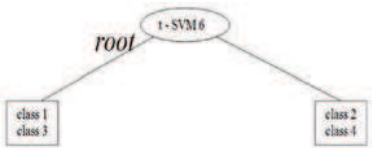
**Rys. 5.** Siedem modeli SVM dzielących zbiór danych na dwa podzbiory
**Fig. 5.** Seven models of SVM that classify data into subsets

**Tab. 1.** Wynik trenowania siedmiu modeli SVM
**Tab. 1.** The result of SVM models training procedure

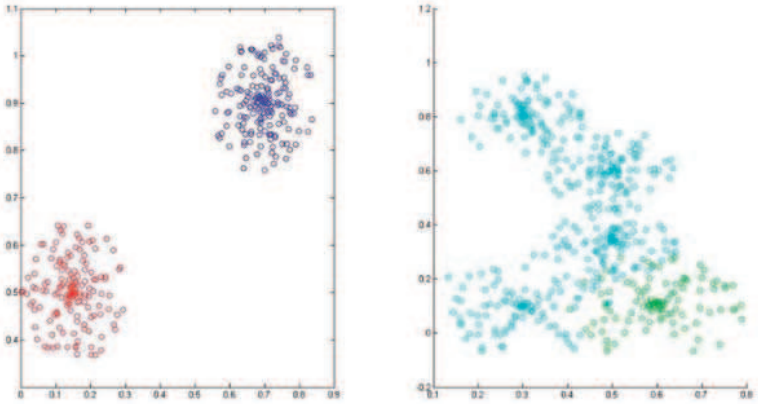
| SVM | Liczba wektorów nośnych (%) | Entropia | Skuteczność klasyfikacji |
|---|---|---|---|
| 1 | 0,6 % | 0,335 | 100 % |
| 2 | 8,24 % | 0,286 | 96,6 % |
| 3 | 0,6 % | 0,317 | 100 % |
| 4 | 7,39 % | 0,416 | 93,09 % |
| 5 | 7,27 % | 0,3 | 97,94 % |
| 6 | 1,33 % | 0,25 | 100 % |
| 7 | 12,6 % | 0,368 | 86,3 % |

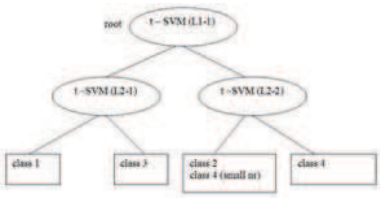
Warto wskazać, że w przypadku modeli SVM1, SVM3 oraz SVM6 otrzymano skuteczność klasyfikacji na poziomie 100 %, co oznacza zerowy błąd klasyfikacji oraz może sugerować nadmierne dopasowanie modeli do zbioru trenującego. W tym przypadku jedynie mniejsza złożoność modelu (mniejsza liczba wektorów nośnych) wpływa na jego wybór. Ostatecznie wygrywającym modelem SVM jest SVM6 ze względu na najmniejszą liczbę wektorów nośnych (1,33 %). Na rys. 6 przedstawiono pierwszy poziom drzewa decyzyjnego z wygrywającym testem SVM6, gdzie w lewej gałęzi znajdują się punkty należące do klas 1 oraz 3, a w prawej gałęzi punkty należące do klas 2 oraz 4.

Ze względu na fakt, że na drugim poziomie drzewa decyzyjnego znajdują się zbiory punktów należących do dwóch różnych klas, obliczono jednego kandydata SVM separującego te

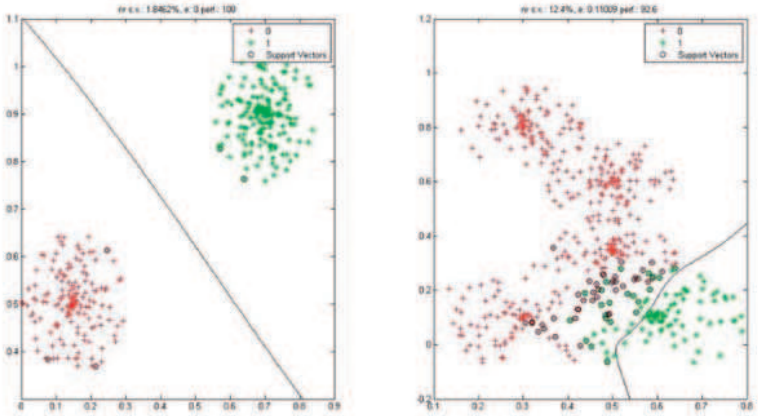



**Rys. 6.** Pierwszy poziom drzew decyzyjnego. Lewa gałąź – punkty należące do klas 1 i 3. Prawa gałąź – punkty należące do klas 2 i 4
**Fig. 6.** First level of the decision tree. Left branch – points of classes 1 and 3. Right branch – points of classes 2 and 4






**Rys. 7.** Drugi poziom drzewa decyzyjnego
**Fig. 7.** Second level of the decision tree



klasy. Warto wskazać, że dla poziomu 2 prawej gałęzi (L2-2) skuteczność otrzymanego modelu wynosi zaledwie 92,6 %, co wymusza przeprowadzenie kolejnych obliczeń.

W celu obliczenia lepszego modelu SVM w gałęzi L2-2, dla trzech kolejnych kandydatów SVM przyjęto parametry zamieszczone w tab. 2.

**Tab. 2.** Parametry SVM
**Tab. 2.** SVM parameters

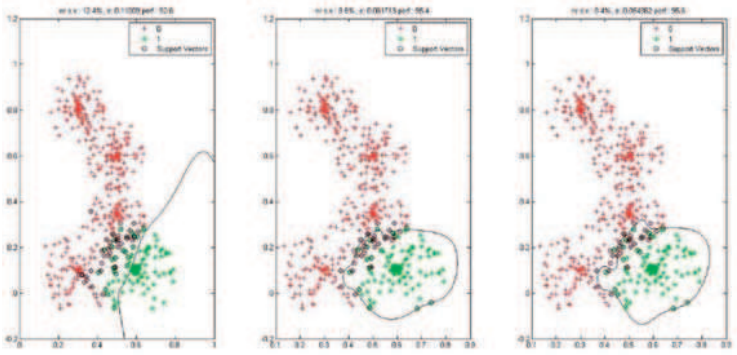
| SVM | KKTViolationLevel | TolKKT |
|---|---|---|
| 1 | 0,1 | 0,1 |
| 2 | 0,05 | 0,05 |
| 3 | 0,01 | 0,01 |

Na rys. 8 przedstawiono granice decyzyjne dla kandydatów SVM z tab. 2.

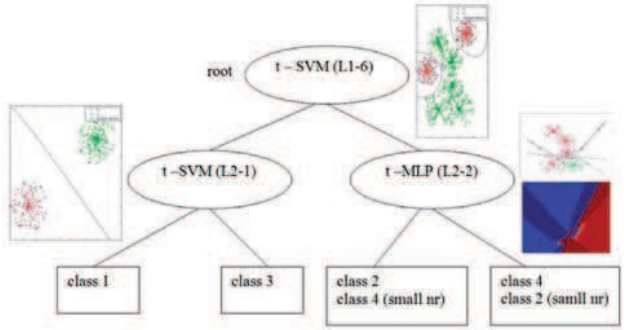
**Tab. 3.** Wynik trenowania trzech kandydatów SVM
**Tab. 3.** SVM training result

| SVM | Liczba wektorów nośnych | Entropia | Skuteczność klasyfikacji |
|---|---|---|---|
| 1 | 12,4 % | 0,11 | 92,6 % |
| 2 | 9,8 % | 0,062 | 95,4 % |
| 3 | 9,4 % | 0,054 | 95,6 % |

Wygrywającym kandydatem dla węzła L2-2 jest SVM3 ze względu na minimalną liczbę wektorów nośnych (9,4 %) oraz jednocześnie największą skuteczność klasyfikacji (95,6 %) przy minimalnej wartości entropii 0,054. Do tego momentu przedstawiono budowę drzewa decyzyjnego na bazie testów – SVM. Dla węzła L2-2 przeprowadzono także eksperyment obliczenia



**Rys. 8.** Modele SVM dla węzła L2-2
**Fig. 8.** SVM model for the node L2-2




**Rys. 9.** Ostateczne drzewo decyzyjne realizujące zadanie klasyfikacji zbioru przykładów należących do czterech klas
**Fig. 9.** The decision tree – a classifier of data belongs to four classes

**Tab. 4.** Wyniki trenowania różnych kandydatów MLP
**Tab. 4.** MLP training result

| MLP | Liczba neuronów w warstwie ukrytej | Entropia | Skuteczność klasyfikacji |
|---|---|---|---|
| 1 | 2 | 0,0224 | 98,2 % |
| 2 | 3 | 0,0339 | 97,2 % |
| 3 | 4 | 0,0228 | 98,6 % |
| 4 | 5 | 0,0245 | 98,2 % |
| 5 | 6 | 0,0092 | 99,6 % |
| 6 | 7 | 0,0169 | 99,0 % |
| 7 | 8 | 0,0000 | 100,0 % |
| 8 | 9 | 0,0109 | 99,2 % |
| 9 | 10 | 0,0009 | 99,8 % |
| 10 | 15 | 0,0000 | 100 % |

optymalnego kandydata spośród MLP o następujących parametrach: liczba epok – 1000, funkcja aktywacji w warstwie ukrytej oraz wyjściowej: *logarithmic sigmoid*, funkcja trenująca *Levenberg-Marquardt backpropagation*, dwa neurony w warstwie wejściowej, jeden neuron w warstwie wyjściowej.

Kandydat MLP5 z sześcioma neuronami w warstwie ukrytej jest optymalnym modelem ze względu na najmniejszą złożoność modelu (liczba neuronów w warstwie ukrytej) oraz najniższą wartość entropii przy akceptowalnej skuteczność klasyfikacji. Dla węzła L2-2 dostępny jest kandydat SVM jak i MLP rozdzielający zbiór danych tego węzła na dwa podzbiory. W związku z tym, że złożoność modelu SVM jest większa od złożoności modelu MLP, wygrywającym kandydatem jest klasyfikator MLP składający się z sześciu neuronów w warstwie ukrytej. W rezultacie drzewo decyzyjne przyjmuje postać z rys. 9.

## Podsumowanie i wnioski

W pracy przedstawiono algorytm drzewa decyzyjnego realizującego hierarchiczną reprezentację wiedzy. Przez zastosowanie klasyfikatorów typu SVM oraz MLP otrzymano nowy zbiór dostępnych testów, a w efekcie końcowym strukturę reprezentacji wiedzy, która:

- w sposób automatyczny tworzy złożone reguły decyzyjne, przez co głębokość drzewa jest znacznie zredukowana
- wykorzystuje zalety algorytmów SVM oraz MLP do tworzenia złożonych granic decyzyjnych separujących dane, które nie są separowalne liniowo
- w sposób automatyczny dokonuje oceny, który z klasyfikatorów SVM czy MLP jest bardziej odpowiedni dla danego poziomu drzewa decyzyjnego.

Największym problemem jest znaczna złożoność obliczeniowa procesu trenowania drzewa decyzyjnego. Jednakże przy zastosowaniu technik przetwarzania równoległego możliwe jest znaczne przyśpieszenie tego procesu, ponieważ występujące tu licznie powtarzające się procedury mogą być zrównoleglone. Warto podkreślić, że proponowany algorytm jest przygotowany do wygenerowania oszczędnych modeli klasyfikujących dane na poszczególnych poziomach drzewa decyzyjnego, w związku z czym może pracować jako szybka maszyna wspomagania decyzji na bazie wektora obserwacji.

## Bibliografia


1. Schürmann T.: *Bias Analysis in Entropy Estimation*. J. Phys. A: Math. Gen. 37 (2004) L295-L301.Trans. Roy. Soc. London, Vol. A247, p. 529–551, April 1995. *(references)*
2. Gottlob G., Pichler R., Wei F.: *Bounded Treewidth as a Key to Tractability of Knowledge Representation and Reasoning*. Vortrag: Twenty-First National Conference on Artificial Intelligence, Boston, Massachusetts, USA; 16.07.2006 - 20.07.2006; [in:] "Proceeding of the Twenty-First National Conference on Artificial Intelligence", The AAAI Press, 2006, p. 250–256.
3. Liya Ding: *A Model of Hierarchical Knowledge Representation – Towards Knowware for Intelligent Systems*. Journal of Advanced Computational Intelligence and Intelligent Informatics Vol. 11 No.1, 2007, p. 1232–1240.
4. de Mingo L.F., Arroyo F., Castellanos J.: *Hierarchical Knowledge Representation to Approximate Functions.* coginf, vol. 1, p.261-267, 2006, 5th IEEE International Conference on Cognitive Informatics Vol. 1, 2006.
5. Min-Yuh Day, Richard Tzong-Han Tsai, Cheng-Lung Sung, Chiu-Chen Hsieh, Cheng-Wei Lee, Shih-Hung Wu, Kun-Pin Wu, Chorng-Shyong Ong, Wen-Lian Hsu: *Reference metadata extraction using a hierarchical knowledge representation framework*. Decision Support Systems, Vol. 43, Issue 1, February 2007, p. 152–167, Mobile Commerce: Strategies, Technologies and Applications.
6. Mohammed I. Bu-Hulatg, Amiya K. Chakravarty: *An object-oriented knowledge representation for hierarchical real-time control of flexible manufacturing.* International Journal of Production Research, Vol. 26, Issue 5 May 1988, p. 777–793.
7. Gentil S. Montmain J.: *Hierarchical representation of complex systems for supporting human decision making*. Advanced Engineering Informatics Vol. 18, Issue 3, July 2004, p. 143–159.
8. Bharadwaj K., Silva J.: *Towards integrating hierarchical censored production rule (hcpr) based system and neural networks*. Lecture Notes Artif. Intelli., 1998, Vol. 1515: p. 121–130.
9. Gilbert J.: *Case-based reasoning applied to instruction method selection for intelligent tutoring systems*. Proceeding of the ITS '2000 Workshop on Case-based Reasoning in Intelligent Training Systems, 2000, p. 11–15.
10. Jones D.M., Paton R.C.: *Toward principles for the representation of hierarchical knowledge in formal ontologies*. Data and Knowledge Engineering, Vol. 31, Nr 2, September 1999. p. 99–113 (15).
11. Hongjun Pan, Jigui Sun: *Complex Knowledge System Modeling Based on Hierarchical Fuzzy Petri Net*. Proceedings of the 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent, 2007, p. 31–34.
12. Ghassaq S. Mosa, Abduladhem A. Ali: *Arabic Phoneme Recognition using Hierarchical Neural Fuzzy Petri Net and LPC Feature Extraction*, Signal Processing: An International Journal (SPIJ) Vol. 3: Issue (5), p. 161–171.
13. Jankowski S., Dusza J. J., Wierzbowski M., Oręziak A.: *SVM Detection of Premature Ectopic Excitations Based on Modified PCA*. 6th International Symposium on Biomedical Data Analysis ISBMDA 2005, (eds. J. L. Oliveiro, V. Maojo, F. Martin-Sanchez, A. Sousa Pereira) Aveiro, Portugalia, Lecture Notes in Bioinformatics LNBI 3745, Springer-Verlag, Berlin Heidelberg, 2005, p. 173–183.
14. Jankowski S., Szymański Z., Piątkowska-Janko E., Oręziak A.: *Improved recognition of sustained ventricular tachycardia from SAECG by support vector machine*. Anatolian Journal of Cardiology, Vol. 7, July, 2007. p. 112–115.
15. Smola J., Schölkopf B.: A Tutorial on Support Vector Regression. NeuroCOLT Technical Report NC-TR-98-030, Royal Holloway College, University of London, UK, 1998.
16. Cichosz P.: *Systemy uczące się*. Wydawnictwo Nukowo-Techniczne Warszawa, 2000.


## Hierarchical knowledge representation for automated reasoning


In the paper the study of knowledge hierarchical representation for automated reasoning is presented. The hierarchical knowledge representation is proposed for predictive modeling purpose. It is improved an effective automated reasoning structure for data set analyzes and making decisions based on complex relations between this data. It is important to emphasize that it is not considered a – priori knowledge concerning data structure, therefore the approach automatically discovers particular constraints between data. It provides a technique of the verification the hierarchical knowledge representation building process that can be useful for the model justification. The presented numerical experiment shows an advantage of proposed approach. It is assumed that the presented automated reasoning can be used for classification purpose where there is a difficulty of proper classifier choice.



**Keywords:** neural network, support vector machine, decision tree, hierarchical knowledge representation



**dr inż. Janusz Będkowski**



Jest pracownikiem naukowym Instytutu Automatyki i Robotyki Politechniki Warszawskiej, Przemysłowego Instytutu Automatyki i Pomiarów oraz inżynierem w Instytucie Maszyn Matematycznych. Obszar badań naukowych to zastosowanie inteligentnych systemów mobilnych, wielorobotowych systemów inspekcyjno-interwencyjnych, projektowanie systemów e-Treningu w zaawansowanej robotyce mobilnej z zastosowaniem technik AR (*Augmented Reality*).

e-mail: jbedkowski@piap.pl

**prof. dr hab. inż. Andrzej Masłowski**



Od 1987 r. profesor w dyscyplinie automatyka i robotyka, jest obecnie profesorem zwyczajnym Politechniki Warszawskiej w Instytucie Automatyki i Robotyki na Wydziale Mechatroniki, a także profesorem w Przemysłowym Instytucie Automatyki i Pomiarów oraz Instytucie Maszyn Matematycznych. Obszar badań naukowych to zastosowanie inteligentnych systemów mobilnych, autonomicznych i zdalnie sterowanych platform mobilnych, wielorobotowych systemów inspekcyjno-interwencyjnych (RISE), projektowanie systemów e-Treningu w zaawansowanej robotyce mobilnej. Jest członkiem IFIP, IFAC, IMACS oraz IMEKO TC-17 Measurement and Control in Robotics. Od 2006 r. przedstawiciel Polski w Joint Coordinating Forum International Advanced Robotics Programme.

e-mail: a.maslowski@mchtr.pw.edu.pl