Raport Badawczy
Research Report

RB/22/2010

Analiza skupień szeregów czasowych

M. Krawczak, G. Szkatuła

Instytut Badań Systemowych
Polska Akademia Nauk

Systems Research Institute
Polish Academy of Sciences

POLSKA AKADEMIA NAUK

Instytut Badań Systemowych

ul. Newelska 6

01-447 Warszawa

tel.: (+48) (22) 3810100

fax: (+48) (22) 3810105

Kierownik Pracowni zgłaszający pracę:
Prof. dr hab. inż. Janusz Kacprzyk

Warszawa 2010

Analiza skupień szeregów czasowych

Maciej Krawczak, Grażyna Szkatuła

Spis treści:

Analiza skupień szeregów czasowych

Wprowadzenie

W ciągu ostatnich kilku lat szeregi czasowe zaczęły stanowić coraz bardziej rozpowszechniony typ danych. Można wyróżnić dwa główne nurty badań z nimi związane:

1) przewidywanie przyszłych zachowań na podstawie dotychczasowych zachowań szeregów,
2) opis danych szeregów czasowych (wyjaśnienie), który może być stosowany do uogólniania, grupowania lub do ich klasyfikacji.

Stosunkowo mało miejsca w literaturze zostało poświęcone symbolicznej reprezentacji szeregów czasowych. Taki stan badań może wydawać się trochę niezrozumiały, a to dlatego, że istnieje wiele efektywnych algorytmów przeznaczonych do analizy długich łańcuchów symboli. Algorytmy takie znalazły zastosowanie w analizie tekstów, a ostatnio w bioinformatyce (Apostolico i in., 2002), Gionis i Mannila, 2003), (Lin i in., 2007). Wydaje się, że można dość łatwo wytłumaczyć taki stan badań, mianowicie stosowane reprezentacje szeregów czasowych oparte na pomiarze odległości pozwalają w dość łatwy sposób porównywać szeregi i określać odległości między nimi. W istniejących algorytmach symbolicznej reprezentacji szeregów czasowych brak jest metod pozwalających na wyliczenie odległości w przestrzeni symbolicznej.

Algorytmy grupowania (analizy skupień) można podzielić się na kilka podstawowych grup:

1) Metody hierarchiczne

- Algorytm zaczyna od takiego podziału zbioru przykładów, w którym każdy przykład stanowi samodzielne skupienie, następnie w kolejnych krokach łączy w skupienia przykłady najbardziej do siebie podobne, a kończy na podziale, w którym wszystkie przykłady należą do jednego skupienia.
- Algorytm zaczyna od skupienia obejmującego wszystkie przykłady, a następnie w kolejnych krokach dzieli je na mniejsze skupienia aż do momentu, gdy każdy przykład stanowi samodzielne skupienie.

2) Metody aglomeracyjne, w której grupowanie polega na wstępnym podzieleniu zbioru przykładów na z góry założoną liczbę grup. Następnie uzyskany podział jest poprawiany w ten sposób, że niektóre przykłady są przenoszone do innych grup, tak, aby uzyskać minimalną wariancję wewnątrz uzyskanych grup.

3) Metody rozmytej analizy skupień (ang. *fuzzy clustering*), które mogą przydzielać przykłady do więcej niż jednej grupy.

W raporcie przedstawiono:

1) Metodykę redukcji wymiarowości szeregów czasowych.

Zaproponowano nowy rodzaj reprezentacji oryginalnych szeregów czasowych o znacznie zmniejszonej wymiarowości, przy zachowaniu zasadniczych cech tych szeregów. Szeregi opisane zostają za pomocą zbioru cech istotnych, reprezentujących zagregowane obwiednie górne lub obwiednie dolne oryginalnych szeregów czasowych, utworzonych przy zastosowaniu pamięci asocjacyjnej, realizowanej przez sieć neuronową.

2) Metodykę grupowania szeregów czasowych w klasy.

Do grupowania przykładów (tworzenia skupień) zaproponowano metodę RG1, która zaczyna od takiego podziału zbioru przykładów, w którym każdy przykład stanowi samodzielną klasę, następnie w kolejnych krokach łączy w skupienia przykłady najbardziej do siebie podobne, a kończy na podziale, w którym wszystkie przykłady należą do jednej klasy. Jest to metodyka zbliżona koncepcyjnie do metod hierarchicznych grupowania danych.

3) Generowanie opisu utworzonych klas w postaci wzorców.

Obliczenia testowe wykonano na danych dostępnych poprzez Internet w bazie danych Uniwersytetu Irvine w Kalifornii, które są często stosowane przy testowaniu algorytmów do eksploracji danych.

Do wygenerowania cech istotnych wybrano program Java Neural Networks Simulator (JavaNNS).

Do tworzenia opisu klas zastosowano metodę IP1, wykorzystującą modyfikację zadania pokrycia zbioru, opisaną przez Szkatułę (1995, 2002), jej modyfikację IP2 oraz G1 - tworzącą minimalny zbiór reguł pewnych dla danej klasyfikacji oraz jej modyfikację G2.

Dokładność klasyfikacji sprawdzano dla szeregów czasowych ze zbioru testowego, który zawierał szeregi, nie biorące udziału w procesie tworzenia opisu klas.

1. Opis zaproponowanego podejścia

Przyjmijmy, że dany jest zbiór szeregów czasowych. Zakładamy, że rozpatrywany n-ty szereg lub jego fragment, $n = 1,2,\ldots,N$, opisany jest wektorem składającym się z K elementów

$$\{x_k(n)\}_{k=1}^{k=K} = [x_1(n), x_2(n), \ldots, x_K(n)]^T \tag{1}$$

dla $k = 1,2,\ldots,K$.

Tworzenie zagregowanych obwiedni górnych i obwiedni dolnych oryginalnych szeregów czasowych

Dla każdego szeregu czasowego (1) tworzone są dwie obwiednie, każda opisana wektorem składającym się z $\left\lfloor \frac{K}{m} \right\rfloor$ elementów, gdzie $m << K$:

- zagregowana m-krokowa obwiednia górna, ozn. $\left\{\bar{x}_k(n)\right\}_{k=1}^{k=\left\lfloor \frac{K}{m} \right\rfloor} = \left[\bar{x}_1(n), \bar{x}_2(n), \ldots, \bar{x}_{\left\lfloor \frac{K}{m} \right\rfloor}(n)\right]^T$,

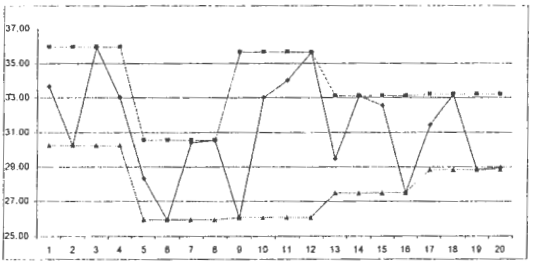
$$\bar{x}_k(n) = \max_{j=m(k-1)+1,\ldots,mk}\{(x_j(n)\}, \tag{2}$$

- zagregowana m-krokowa obwiednia dolna, ozn. $\left\{\underline{x}_k(n)\right\}_{k=1}^{k=\left\lfloor \frac{K}{m} \right\rfloor} = \left[\underline{x}_1(n), \underline{x}_2(n), \ldots, \underline{x}_{\left\lfloor \frac{K}{m} \right\rfloor}(n)\right]^T$,

$$\underline{x}_k(n) = \min_{j=m(k-1)+1,\ldots,mk}\{(x_j(n)\} \tag{3}$$

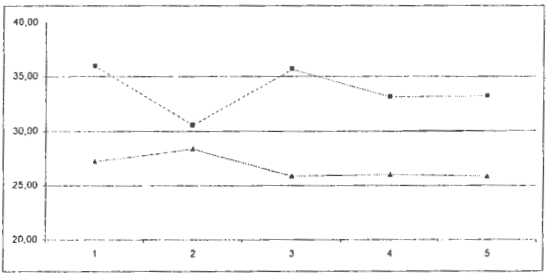
dla $k = 1,2,\ldots,\left\lfloor \frac{K}{m} \right\rfloor$.

Nowa reprezentacja szeregu czasowego oparta na koncepcji obwiedni górnych i dolnych została zilustrowana na rys. 1 i rys. 2.



Rys. 1. *4*-krokowa obwiednia górna i dolna przykładowego szeregu czasowego – przed agregacją

Zagregowane *4*-krokowe obwiednie górna i dolna – odpowiadające przykładowi z rys. 1 przedstawione są na Rys. 2.



Rys. 2. Zagregowane *4*-krokowe obwiednie górna i dolna

Generowanie cech istotnych

Generowanie cech istotnych jest ściśle związane z problemem kompresji danych lub redukcją wymiarowości szeregów czasowych. Zadaniem kompresji danych jest takie zmniejszenie informacji o szeregu czasowym, aby można było odtworzyć ten szereg. Jest to dekompresja danych, przy możliwie małych stratach informacji w stosunku do informacji oryginalnej.

Rozpatrzmy szereg lub jego fragment zapisany w postaci wektora składającego się z N elementów postaci

$$\bar{x} = \left\{ \bar{x}_k(n) \right\}_{k=1}^{k=\left\lfloor \frac{K}{m} \right\rfloor} = \left[\bar{x}_1(n), \bar{x}_2(n), \ldots, \bar{x}_{\left\lfloor \frac{K}{m} \right\rfloor}(n) \right]^T \tag{4a}$$

reprezentujący obwiednie górne szeregów czasowych (1), lub

$$x = \left\{ x_k(n) \right\}_{k=1}^{k=\left\lfloor \frac{K}{m} \right\rfloor} = \left[x_1(n), x_2(n), \ldots, x_{\left\lfloor \frac{K}{m} \right\rfloor}(n) \right]^T \tag{4b}$$

reprezentujący obwiednie dolne szeregów czasowych (1).

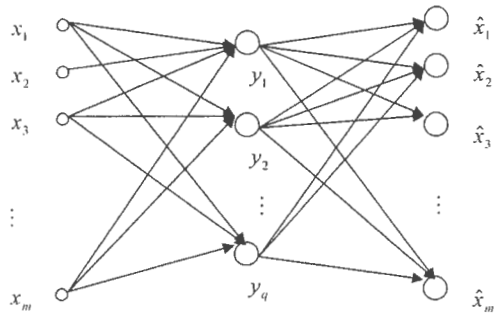
Po kompresji szereg (4a) lub (4b) jest reprezentowany przez q cech istotnych, tzn. przez następujący wektor

$$y = \left\{ y_i(n) \right\}_{i=1}^{i=q} = \left[y_1(n), y_2(n), \ldots, y_q(n) \right]^T, \tag{5}$$

gdzie $q << \left\lfloor \frac{K}{m} \right\rfloor$. Zakłada się, że na podstawie wektora y można odtworzyć wektor x lub x z pewną dokładnością.

Szereg czasowy jest teraz reprezentowany przez wektor y, którego elementy tworzą cechy istotne, inaczej składniki główne (principal components) (Oja, 1992).

Kompresja zagregowanych m-krokowych obwiedni (wzór (2) i (3)) dokonywana jest z zastosowaniem pamięci heteroasocjacyjnej (realizowanej poprzez jednokierunkową sieć neuronową z jedną warstwą ukrytą, którą stanowi q neuronów, gdzie $q << \left\lfloor \frac{K}{m} \right\rfloor$).



Rysunek 3. Zaproponowana struktura sieci

Zagregowane m-krokowe obwiednie są reprezentowane teraz przez zbiór wartości q cech istotnych. Dla wybranej permutacji liczb (1, 2, ..., q) możemy opisać daną zagregowaną m-krokową obwiednię górną w postaci zbioru,

$$\{\bar{y}_j(n)\}_{j=1}^{j=q} = \{\bar{y}_1(n), \bar{y}_2(n), \ldots, \bar{y}_q(n)\} \tag{6}$$

lub

$$\{\overset{1}{\bar{y}}_1(n), \overset{2}{\bar{y}}_2(n), \ldots, \overset{q}{\bar{y}}_q(n)\} \tag{6a}$$

oraz daną zagregowaną m-krokową obwiednię dolną w postaci zbioru

$$\{\underline{y}_j(n)\}_{j=1}^{j=q} = \{\underline{y}_1(n), \underline{y}_2(n), \ldots, \underline{y}_q(n)\} \tag{7}$$

lub

$$\{\overset{1}{\underline{y}}_1(n), \overset{2}{\underline{y}}_2(n), \ldots, \overset{q}{\underline{y}}_q(n)\} \tag{7a}$$

dla $n = 1, 2, \ldots, N$.

Przekształcanie cech istotnych

Oryginalny zbiór cech opisujących przykłady (szeregi czasowe) można modyfikować, usuwając niektóre z nich, dodając nowe lub też zastępując istniejące cechy nowymi. Zmiana zbioru cech opisujących przykłady może w istotny sposób wpływać na sposób tworzenia skupień jak również na postać tych skupień. Oczywiste jest, że żadne przekształcenia nie potrafią stworzyć nowej informacji, której nie ma w wartościach cech opisujących przykłady, wprost przeciwnie, można stracić część informacji, która była zawarta w oryginalnym zbiorze cech, nigdy na odwrót.

W rozpatrywanym w pracy zagadnieniu, wygenerowane cechy istotne opisujące przykłady zostały zastąpione nowymi cechami, zawierającymi w sobie informacje o różnicach w wartościach cech dla danego przykładu. Wprowadzenie nowych cech istotnych zwiększa wymiarowość problemu, ale jednocześnie pozwala wprowadzić pojęcie odległości pomiędzy przykładami konieczne do analizy grupowania przykładów. Nowe cechy nie pozwalają opisywać przykładów w jakimkolwiek stopniu dokładniej, ale pozwalają czasem opisać prościej, w sposób mniej złożony. Tworzone skupienia mogą mieć prostszy opis uzyskany za pomocą nowych cech niż za pomocą oryginalnych. Przekształcenie zbioru cech umożliwia w pewnym sensie uwzględnienie naszych preferencji przy określaniu podobieństwa przykładów, które chcemy, aby były uwzględniane podczas grupowania.

Nowe cechy istotne zostały wprowadzone w następujący sposób. Rozpatrzono wszystkie dwuelementowe kombinacje bez powtórzeń zbioru {1, 2, ..., q} cech oryginalnych, tzn. $\binom{q}{2}$ dwuelementowych kombinacji postaci: {1,2},...,{1,q}, {2,3}, ..., {2,q},...,{q-1, q}. Odpowiadające im różnice w wartościach cech istotnych tworzą odpowiednio nowy zbiór cech (który stanowi zbiór różnic), które można w przypadku uwzględniania zagregowanych obwiedni górnych zapisać w postaci:

$$\{r_j(n)\}_{j=1}^{j=\binom{q}{2}} = \{y_1(n)-y_2(n),...,y_1(n)-y_q(n),...,y_2(n)-y_2(n),...,y_2(n)-y_q(n),...,y_{q-1}(n)-y_q(n)\} \tag{8}$$

lub

$$\{\underset{(1,2)}{y_1(n)-y_2(n)},\ldots,\underset{(1,q)}{y_1(n)-y_q(n)},\underset{(2,3)}{y_2(n)-y_3(n)},\ldots,\underset{(2,q)}{y_2(n)-y_q(n)},\ldots,\underset{(q-1,q)}{y_{q-1}(n)-y_q(n)}\} \tag{8a}$$

a dla zagregowanych obwiedni dolnych w postaci:

$$\{r_j(n)\}_{j=1}^{j=\binom{q}{2}} = \{y_1(n)-y_2(n),...,y_1(n)-y_q(n),y_2(n)-y_3(n),...,y_2(n)-y_q(n),...,y_{q-1}(n)-y_q(n)\} \tag{9}$$

lub

$$\{\underset{(1,2)}{y_1(n)-y_2(n)},\ldots,\underset{(1,q)}{y_1(n)-y_q(n)},\underset{(2,3)}{y_2(n)-y_3(n)},\ldots,\underset{(2,q)}{y_2(n)-y_q(n)},\ldots,\underset{(q-1,q)}{y_{q-1}(n)-y_q(n)}\} \tag{9a}$$

Tak więc, szeregi czasowe zostają opisane $\binom{q}{2}$ nowymi cechami.

Dla uproszczenia zapisu, w dalszej części pracy będą stosowane uproszczone oznaczenia dla zbioru nowych cech, $\{r_j\}_{j=1}^{j=\binom{q}{2}}$, jeśli nie będzie to prowadziło do niejednoznaczności.

Nominalizacja wartości cech

Nowe cechy opisujące przykłady są następnie nominalizowane, tzn. zastąpione cechami o wartościach dyskretnych, odpowiadających pewnym przedziałom ciągłych wartości oryginalnej cechy.

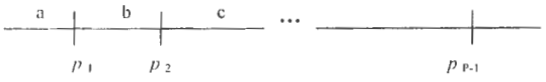
Niech cecha $r_j \in A_r$, $j = 1,...,\binom{q}{2}$, zbiór $V_{r_j} = \{v_{j,1}, v_{j,2},..., v_{j,L_j}\}$ jest dziedziną cechy r_j, L_j oznacza liczbę wartości j-tej cechy. Można określić podprzedziały przedziału zmienności wartości dla wszystkich cech, określając punkty cięcia w następujący sposób:

$$p_i = \min\{V_{r_1}, V_{r_2},...V_{r_{\binom{q}{2}}}\} + i \cdot \partial, \tag{10}$$

$$\partial = \frac{\max\{V_{r_1}, V_{r_2},...V_{r_{\binom{q}{2}}}\} - \min\{V_{r_1}, V_{r_2},...V_{r_{\binom{q}{2}}}\}}{P},$$

gdzie $i = 1,..., P-1$, $P \in \mathrm{N}$ jest parametrem podziału.

Poszczególnym podprzedziałom w ramach przedziału zmienności cech możemy przyporządkować odpowiednie symbole, np. kolejne litery alfabetu, patrz rys.4.



Rys. 4. Podział przedziału zmienności cech istotnych.

W ten sposób określamy symboliczną reprezentację cech istotnych, a tym samym symboliczną reprezentację szeregów czasowych. Tak więc każdy przykład możemy teraz zapisać w postaci nominalnej, np. w postaci ciągu symboli (b, c, e, a,...,a), gdzie symbole odpowiadają przyjętym przedziałom ciągłych wartości dla wszystkich cech.

Przedstawiony powyżej równomierny podział przedziału zmienności wartości wszystkich cech wydaje się być naturalny i prosty, jednak nie uwzględnia częstości występowania wartości cech w przykładach. Wydaje się, że można również rozważać symboliczną reprezentację klas szeregów czasowych uwzględniając równe prawdopodobieństwo występowania symboli. Dla symbolicznej reprezentacji szeregów czasowych otrzymywanej

bezpośrednio z wartości szeregów czasowych zagadnienie dyskretyzacji przedziału zmienności w taki sposób, aby występowanie wartości symboli miało takie same prawdopodobieństwo było rozważane np. w pracach Apostolico i in. (2002) oraz Lin i in. (2007).

Zadanie grupowania

Metody grupowania danych, inaczej analizy skupień, są technikami eksploracji danych i należą do zagadnień uczenia maszynowego na podstawie przykładów, które nie wymagają nadzorowania. Zadanie grupowania polega na znajdowaniu ukrytych wzorców w danych źródłowych. Analiza skupień szczególnie dobrze nadaje się do analizowania punktów w przestrzeni wielowymiarowej, gdy zdolności analityczne człowieka nie są w stanie objąć tak szerokiej złożoności ukrytej zależności między przykładami.

Metody grupowania służą do znajdowania wśród danych wejściowych podzbiorów zawierających możliwie podobne do siebie przykłady. Podobieństwo pomiędzy przykładami jest z reguły wyrażane na podstawie określonej funkcji metryki, czyli odległości między przykładami. Podobieństwo między przykładami charakteryzowane jest niewielką wzajemną odległością. Podzbiory rozpatrywanego zbioru przykładów nazywane są klastrami lub gronami.

Załóżmy, że dany jest zbiór przykładów $U = \{e^n\}$, $n = 1, 2, \ldots, N$. Przykłady te opisujemy za pomocą K cech $A = \{a_1, \ldots, a_K\}$ o skończonych zbiorach wartości, odpowiednio $V_{a_j} = \{v_{j,1}, v_{j,2}, \ldots, v_{j,L_j}\}$ dla $a_j \in A$, $j = 1, \ldots, K$, L_j - określa liczbę wartości j-tej cechy.

Każdy przykład $e^n \in U$ można opisać za pomocą koniunkcji K warunków elementarnych w postaci

$$e^n = (a_1 = v_{1,t(1,n)}) \wedge \ldots \wedge (a_K = v_{K,t(K,n)})$$

gdzie $v_{j,t(j,n)} \in V_{a_j}$, $j = 1, \ldots, K$. Indeks $t(j,n)$ dla $j \in \{1, 2, \ldots, K\}$ oraz $n \in \{1, 2, \ldots, N\}$ określa, którą wartość przyjmuje j-ta cecha w n-tym przykładzie.

Proces grupowania polega na podziale danego zbioru przykładów U na c grup $\{U_1, U_2, \ldots, U_c\}$.

Niech dla danej cechy a_j zbiory $A_{j,t(j,k)}$ i $A_{j,t(j,n)}$ będą dowolnymi, nie pustymi podzbiorami zbioru wartości tej cechy, tzn. $A_{j,t(j,k)} \subseteq V_{a_j}$, $A_{j,t(j,n)} \subseteq V_{a_j}$

Def.
Mówimy, że warunek $(a_j \in A_{j,t(j,k)})$ *dominuje warunek* $(a_j \in A_{j,t(j,n)})$ jeżeli $A_{j,t(j,k)} \supseteq A_{j,t(j,n)}$, ozn. $(a_j \in A_{j,t(j,k)}) \succ (a_j \in A_{j,t(j,n)})$.

Np. Warunek $(a_j \in \{a,b,f\})$ dominuje warunek $(a_j \in \{a,f\})$, co oznaczamy $(a_j \in \{a,b,f\}) \succ (a_j \in \{a,f\})$.

Mówimy o wzajemnym braku dominacji dwóch warunków, jeśli warunek $(a_j \in A_{j,t(j,k)})$ *nie dominuje* warunku $(a_i \in A_{i,t(i,n)})$, ani warunek $(a_i \in A_{i,t(i,n)})$ nie dominuje warunku $(a_j \in A_{j,t(j,k)})$, sytuację taką oznaczamy $(a_j \in A_{j,t(j,k)}) \prec\succ (a_i \in A_{i,t(i,n)})$.

Np. Warunek $(a_j \in \{a,b,f\})$ nie dominuje warunku $(a_j \in \{a,c\})$, ani warunek $(a_j \in \{a,c\})$ nie dominuje warunku $(a_j \in \{a,b,f\})$, co oznaczamy $(a_j \in \{a,b,f\}) \prec\succ (a_j \in \{a,c\})$.

Def.
Grupą U_g zapisujemy w postaci $(a_1 = A_{1,t(1,g)}) \wedge ... \wedge (a_K = A_{K,t(K,g)})$, gdzie $A_{j,t(j,g)} \subseteq V_{a_j}$ dla $j = 1,...,K$.

Mówimy, że przykład $e^n = (a_1 = v_{1,t(1,n)}) \wedge ... \wedge (a_K = v_{K,t(K,n)})$ *należy do grupy* U_g jeżeli spełnione są zależności:

$$(a_1 \in A_{1,t(1,g)}) \succ (a_1 = v_{1,t(1,n)}), \ldots, (a_K \in A_{K,t(K,g)}) \succ (a_K = v_{K,t(K,n)}) \qquad (11)$$

Tak więc, do danej grupy U_g należą przykłady $\{e^n : e^n \in U, n \in I_g \subseteq \{1,2,...,N\}\}$ spełniające warunek (11).

Rozważmy grupę U_{g_1} opisaną w postaci $(a_1 \in A_{1,t(1,g_1)}) \wedge ... \wedge (a_K \in A_{K,t(K,g_1)})$ oraz grupę U_{g_2} opisaną w postaci $(a_1 \in A_{1,t(1,g_2)}) \wedge ... \wedge (a_K \in A_{K,t(K,g_2)})$, $A_{j,t(j,n)} \subseteq V_{a_j}$, $A_{j,t(j,m)} \in V_{a_j}$ dla $j = 1,...,K$.

Def.
Mówimy, że grupy g_1 i g_2 są *ω-rozróżnialne* na zbiorze cech $\{a_j : j \in I_k\}$, $card(I_k) = \omega$ jeśli zachodzą warunki:

1) Dokładnie dla ω cech zachodzi brak wzajemnej dominacji, tzn.
$(a_j = A_{j,t(j,g_1)}) \prec\succ (a_j = A_{j,t(j,g_2)})$ dla $a_j \in I_k$, $card(I_k) = \omega$
2) $(a_j = A_{j,t(j,g_1)}) \succ (a_j = A_{j,t(j,g_2)})$ lub $(a_j = A_{j,t(j,g_2)}) \succ (a_j = A_{j,t(j,g_1)})$ dla $\forall j \in \{1,...,K\} \setminus I_k$

Def.

Jeśli dwie grupy g_1 i g_2 są ω-rozróżnialne na zbiorze cech $\{a_j : j \in I_k\}$, $card(I_k) = \omega$, to możemy określić ω - *warunkową regułę akcji* tworzącą nową grupę g_3 w postaci

$$\bigwedge_{j \in I_k} (A_{j,i(j,g_1)} := A_{j,i(j,g_1)} \cup A_{j,i(j,g_2)}) \bigwedge_{j \in \{1,2,\ldots K\} \setminus I_k} (A_{j,i(j,g_3)} := dom\{A_{j,i(j,g_1)}, A_{j,i(j,g_2)}\})$$

$$\Rightarrow ((g_1, g_2) \rightarrow (g_3)) \qquad (12)$$

gdzie *dom* - dominujący warunek z pary warunków.

Przykład.

Z połączenia dwóch 4 – rozróżnialnych grup przykładów g_1 i g_2 na zbiorze cech $I_k = \{1,4,5,10\}$, przedstawionych poniżej

(w przyjętej notacji, w kolumnie 1 zamieszczone są numery przykładów należących do danej grupy, w kolejnych kolumnach wartości cech)

Grupy \ Cechy	r1	r2	r3	r4	r5	r6	r7	r8	r9	r10
g_1: 57, 51	e	e	g ∨ h	e ∨ d	d	g	g	f ∨ g	f	f
g_2: 55, 60, 63	f	e	g	f ∨ e	e	g	g	g	g ∨ f	g

uzyskujemy nową grupę g_3, zawierającą przykłady z obu grup, stosując regułę (12)

Grupa	r1	r2	r3	r4	r5	r6	r7	r8	r9	r10
g_3: 57, 51, 55, 60, 63	e ∨ f	e	g ∨ h	e ∨ d ∨ f	d ∨ e	g	g	f ∨ g	f ∨ g	f ∨ g

Zaproponowany algorytm grupowania przykładów (tworzenia skupień) ozn. RG1, przedstawiono poniżej:

Krok 1.

Dany jest zbiór przykładów $U = \{e^n\}$, $n = 1,2,\ldots,N$.

K – liczba cech opisujących przykłady, LK - przyjęta liczba grup, $\omega = 0$

Każdy przykład tworzy jednoelementową grupę w G, $card(G) = N$.

Krok 2.

- Każdą parę ω - rozróżnialnych grup w G łączymy w jedną grupę, wzór (11), zawierającą przykłady z obu grup.
- Jeśli $card(G) = LK$ przechodzimy do kroku 3; w przeciwnym przypadku $\omega := \omega + 1$ i jeśli $\omega \leq K$ powtarzamy krok 2; a w przeciwnym przypadku przechodzimy do kroku 3.

Krok 3. STOP.

2. Przykład obliczeniowy 1

Zaproponowaną metodę grupowania RG1 dokładniej omówiono poniżej na przykładzie.

Ze zbioru sztucznie wygenerowanych szeregów czasowych opisanych w rozdz. 3 do obliczeń wybrano 21 przykładów: 7 przykładów z klasy 1, 7 z klasy 2 oraz 7 z klasy 3. Przykłady zostały opisane za pomocą zbioru cech r_i dla $i = 1,...,10$. Informacja o przynależności przykładu do klasy służyła wyłącznie do równomiernego wyboru przykładów z klas, a nie była brana pod uwagę przy ich grupowaniu.

Na starcie utworzono 21 grup przedstawionych poniżej, każdy przykład ze zbioru utworzył grupę. W przyjętej notacji, w kolumnie 1 zamieszczone są numery przykładów należących do danej grupy, w kolejnych kolumnach wartości cech.

Krok 1

$K = 10$
$LK = 3$
$\omega = 0$
Zbiór G zawiera 21 grup jednoelementowych przedstawionych poniżej.

Grupy	r1	r2	r3	r4	r5	r6	r7	r8	r9	r10
25	d	f	f	i	e	f	i	e	i	h
4	d	g	e	i	f	f	h	e	i	h
14	e	e	d	i	e	d	h	d	h	h
22	e	e	d	i	e	d	h	d	h	h
12	f	e	d	j	e	d	h	d	h	h
18	f	e	d	i	e	d	h	d	h	h
24	f	e	d	i	e	d	h	d	h	h
35	e	e	h	b	e	h	d	h	d	d
40	e	e	h	b	e	h	d	h	d	d
49	e	e	h	b	e	h	d	h	d	d
33	f	e	h	a	e	h	d	h	d	d
41	f	e	h	a	e	h	d	h	d	d
38	g	e	h	a	g	g	d	i	e	e
48	h	f	f	b	h	g	e	i	e	e
75	d	g	e	g	f	g	f	e	h	f
61	d	h	e	f	f	g	e	f	h	f
57	e	e	g	e	d	g	g	f	f	f
51	e	e	h	d	d	g	g	g	f	f
55	f	e	g	f	e	g	g	g	g	g
60	f	e	g	e	e	g	g	g	g	g
63	f	e	g	e	e	g	g	g	f	g

Krok 2. $\omega = 0$

Każdą parę ω - rozróżnialnych grup w G łączymy w jedną grupę. Przykłady identycznie opisane zostaną połączone w jedną grupę.

G zawiera 16 grup zamieszczonych poniżej, w tym 4 nowe (przyciemnione).

Grupy	r1	r2	r3	r4	r5	r6	r7	r8	r9	r10
25	d	f	f	i	e	f	i	e	i	h
4	d	g	e	i	f	f	h	e	i	h
14, 22	e	e	d	i	e	d	h	d	h	h
12	f	e	d	j	e	d	h	d	h	h
18, 24	f	e	d	i	e	d	h	d	h	h
35, 40, 49	e	e	h	b	e	h	d	h	d	d
33, 41	f	e	h	a	e	h	d	h	d	d
38	g	e	h	a	g	g	d	i	c	e
48	h	f	f	b	h	g	c	i	c	e
75	d	g	e	g	f	g	f	e	h	f
61	d	h	e	f	f	g	e	f	h	f
57	e	e	g	e	d	g	g	f	f	f
51	e	e	h	d	d	g	g	g	f	f
55	f	e	g	f	e	g	g	g	g	g
60	f	e	g	e	e	g	g	g	g	g
63	f	e	g	e	e	g	g	g	f	g

$\omega = \omega + 1$ i powtarzamy Krok 2.

Krok 2, $\omega = 1$.

Każdą parę ω - rozróżnialnych grup w G łączymy w jedną grupę.

G zawiera 12 grup zamieszczonych poniżej, w tym 2 nowe (przyciemnione).

Grupy	r1	r2	r3	r4	r5	r6	r7	r8	r9	r10
25	d	f	f	i	e	f	i	e	i	h
4	d	g	e	i	f	f	h	e	i	h
14, 22, 12, 18, 24	e∨f	e	d	i∨j	e	d	h	d	h	h
35, 40, 49	e	e	h	b	e	h	d	h	d	d
33, 41	f	e	h	a	e	h	d	h	d	d
38	g	e	h	a	g	g	d	i	c	e
48	h	f	f	b	h	g	c	i	c	e
75	d	g	e	g	f	g	f	e	h	f
61	d	h	e	f	f	g	e	f	h	f
57	e	e	g	e	d	g	g	f	f	f
51	e	e	h	d	d	g	g	g	f	f
55, 60, 63	f	e	g	f∨e	e	g	g	g	g∨f	g

$\omega = \omega + 1$ i powtarzamy Krok 2.

Krok 2, $\omega = 2$

Każdą parę ω - rozróżnialnych grup w G łączymy w jedną grupę.
G zawiera 11 grup zamieszczonych poniżej, w tym 1 nowa (przyciemniona).

Grupy	r1	r2	r3	r4	r5	r6	r7	r8	r9	r10
25	d	f	f	i	e	f	i	e	i	h
4	d	g	e	i	f	f	h	e	i	h
14, 22, 12, 18, 24	e∨f	e	d	i∨j	e	d	h	d	h	h
35, 40, 49, 33, 41	e∨f	e	h	b∨a	e	h	d	h	d	d
38	g	e	h	a	g	g	d	i	c	e
48	h	f	f	b	h	g	c	i	c	e
75	d	g	e	g	f	g	f	e	h	f
61	d	h	e	f	f	g	e	f	h	f
57	e	e	g	e	d	g	g	f	f	f
51	e	e	h	d	d	g	g	g	f	f
55, 60, 63	f	e	g	f∨e	e	g	g	g	g∨f	g

$\omega = \omega + 1$ i powtarzamy Krok 2.

Krok 2 , $\omega = 3$.

Każdą parę ω - rozróżnialnych grup w G łączymy w jedną grupę.

G zawiera 10 grup zamieszczonych poniżej, w tym 1 nowa (przyciemniona)..

Grupy	r1	r2	r3	r4	r5	r6	r7	r8	r9	r10
25	d	f	f	i	e	f	i	e	i	h
4	d	g	e	i	f	f	h	e	i	h
14, 22, 12, 18, 24	e∨f	e	d	i∨j	e	d	h	d	h	h
35, 40, 49, 33, 41	e∨f	e	h	b∨a	e	h	d	h	d	d
38	g	e	h	a	g	g	d	i	c	e
48	h	f	f	b	h	g	c	i	c	e
75	d	g	e	g	f	g	f	e	h	f
61	d	h	e	f	f	g	e	f	h	f
57, 51	e	e	g∨h	e∨d	d	g	g	f∨g	f	f
55, 60, 63	f	e	g	f∨e	e	g	g	g	g∨f	g

$\omega = \omega + 1$ i powtarzamy Krok 2.

Krok 2, $\omega = 4$.

Każdą parę ω - rozróżnialnych grup w G łączymy w jedną grupę.

G zawiera 7 grup zamieszczonych poniżej, w tym 3 nowe (przyciemnione).

Grupy	r1	r2	r3	r4	r5	r6	r7	r8	r9	r10
25, 4	d	f∨g	f∨e	i	e∨f	f	i∨h	e	i	h
14, 22, 12, 18, 24	e∨f	e	d	i∨j	e	d	h	d	h	h
35, 40, 49, 33, 41	e∨f	e	h	b∨a	e	h	d	h	d	d
38	g	e	h	a	g	g	d	i	c	e
48	h	f	f	b	h	g	c	i	c	e
75, 61	d	g∨h	e	g∨f	f	g	f∨e	e∨f	h	f
57, 51, 55, 60, 63	e∨f	e	g∨h	e∨d∨f	d∨e	g	g	f∨g	f∨g	f∨g

$\omega = \omega + 1$ i powtarzamy Krok 2.

Krok 2. $\omega = 5$.

Nie znaleziono ω - rozróżnialnych grup w *G*.

$\omega = \omega + 1$ i powtarzamy Krok 2.

Krok 2, $\omega = 6$.

Każdą parę ω - rozróżnialnych grup w *G* łączymy w jedną grupę.

G zawiera 4 grupy zamieszczone poniżej, w tym 3 nowe (przyciemnione).

Grupy	r1	r2	r3	r4	r5	r6	r7	r8	r9	r10
25, 4,14, 22, 12, 18, 24	d∨e∨f	f∨g∨e	f∨e∨d	i∨j	e∨f	f∨d	i∨h	e∨d	i∨h	h
35, 40, 49, 33, 41, 38, 48	e∨f∨g∨h	e∨f	h∨f	b∨a	e∨g∨h	h∨g	d∨c	h∨i	d∨c	d∨e
75, 61	d	g∨h	e	g∨f	f	g	f∨e	e∨f	h	f
57, 51, 55, 60, 63	e∨f	e	g∨h	e∨d∨f	d∨e	g	g	f∨g	f∨g	f∨g

$\omega = \omega + 1$ i powtarzamy Krok 2.

Krok 2, $\omega = 7$.

Nie znaleziono ω - rozróżnialnych grup w *G*.

$\omega = \omega + 1$ i powtarzamy Krok 2.

Krok 2, $\omega = 8$.

Każdą parę ω - rozróżnialnych grup w *G* łączymy w jedną grupę.

G zawiera 3 grupy zamieszczone poniżej, w tym 1 nowa (przyciemniona).

Grupy	r1	r2	r3	r4	r5	r6	r7	r8	r9	r10
25, 4, 14, 22, 12, 18, 24	d∨e∨f	f∨g∨e	f∨e∨d	i∨j	e∨f	f∨d	i∨h	e∨d	i∨h	h
35, 40, 49, 33, 41, 38, 48	e∨f∨g∨h	e∨f	h∨f	b∨a	e∨g∨h	h∨g	d∨c	h∨i	d∨c	d∨e
75, 61, 57, 51, 55, 60, 63	d∨e∨f	g∨h∨e	e∨g∨h	g∨f∨e∨d	f∨d∨e	g	f∨e∨g	e∨f∨g	h∨f∨g	f∨g

$card(G) = LK$ przechodzimy do Kroku 3.

Krok 3.

Utworzono 3 grupy:

Grupa 1: przykłady 25, 4, 14, 22, 12, 18, 24

Grupa 2: przykłady 35, 40, 49, 33, 41, 38, 48

Grupa 3: przykłady 75, 61, 57, 51, 55, 60, 63

Wygenerowane 3 grupy są całkowicie zgodne z pierwotną przynależnością rozpatrywanych przykładów do klas, nie braną pod uwagę przy ich grupowaniu.

Następnie dla trzech wygenerowanych klas utworzono opisy, przedstawione poniżej:

klasa	r1	r2	r3	r4	r5	r6	r7	r8	r9	r10
1	*	*	*	i∨j	*	*	*	*	*	*
2	*	*	*	a∨b	*	*	*	*	*	*
3	*	*	*	d∨e∨f∨g	*	*	*	*	*	*

które w 100% poprawnie klasyfikują wszystkie szeregi czasowe ze zbioru.

3. Przykład obliczeniowy 2
(4-krokowe obwiednie dolne)

Opis danych

Do obliczeń wybrano zbiór danych dostępny poprzez Internet w bazie danych Uniwersytetu Irvine w Kalifornii, który jest często stosowany przy testowaniu algorytmów do eksploracji danych (Alcock i Manolopoulos, 1999):

http://kdd.ics.uci.edu/databases/synthetic_control/synthetic_control.data.html

Zawiera on sztucznie wygenerowane szeregi czasowe, z których każdy złożony jest z 60 liczb, określona jest też przynależność szeregu do jednej z sześciu klas.

Do obliczeń wybrano trzy klasy:

- klasa 1: szeregi typu E (25 szeregów uczących + 25 szeregów testowych)
- klasa 2: szeregi typu F (25 szeregów uczących + 25 szeregów testowych)
- klasa 3: szeregi typu A (25 szeregów uczących + 25 szeregów testowych)

Rozpatrywane znormalizowane szeregi czasowe ze zbioru uczącego można zapisać w postaci $[x_1(n), x_2(n), ..., x_{60}(n)]^T$, n = 1,2, ... ,75 (wzór(1)). Znana przynależność każdego szeregu do jednej z rozpatrywanych klas nie była uwzględniana w procesie grupowania (tworzenia skupień szeregów). Posłużyła jedynie dla porównania faktycznej przynależności szeregów z wygenerowanymi klasami.

Dla każdego n-tego szeregu uczącego zawierającego 60 wartości liczbowych tworzona była zagregowana 4-krokowa obwiednia górna (wzór (2)),

$$\left\{\bar{x}_k(n)\right\}_{k=1}^{k=\left\lfloor \frac{K}{m} \right\rfloor} = [\bar{x}_1(n), \bar{x}_2(n), ..., \bar{x}_{15}(n)]^T,$$

oraz zagregowana 4-krokowa obwiednia dolna (wzór (3))

$$\left\{\underline{x}_k(n)\right\}_{k=1}^{k=\left\lfloor \frac{K}{m} \right\rfloor} = [\underline{x}_1(n), \underline{x}_2(n), ..., \underline{x}_{15}(n)]^T.$$

dla $n = 1,2,\ldots,75$.

Następnie wygenerowano 5 cech istotnych, stosując w tym celu pamięć asocjacyjną, realizowaną poprzez trójwarstwową jednokierunkową sieć neuronową o 15 wejściach i 15 wyjściach, z jedną warstwą ukrytą, którą stanowiło 5 neuronów ($q = 5$). Do tego celu został użyty program Java Neural Networks Simulator (JavaNNS).

Generowanie q cech istotnych wykonano przyjmując za dane uczące zagregowane 4-krokowe obwiednie dolne (wzór (3)).

Po procesie uczenia sieci, otrzymane dla każdego szeregu wartości w warstwie ukrytej (przemnożone przez 1000) utworzyły skompresowany opis szeregu. Tak więc z zastosowaniem wartości cech istotnych zapisano każdy rozpatrywany szereg czasowy $\{x_k(n)\}_{k=1}^{k=60} = [x_1(n), x_2(n), \ldots, x_{60}(n)]^T$, dla $n = 1,2,\ldots,75$, w postaci zbiorów, odpowiednio:

$$\{y_i(n)\}_{i=1}^{i=5} = \{y_1(n), y_2(n), \ldots, y_5(n)\},\ n = 1,2,\ldots,75 \text{ (wzór (5)).}$$

Wygenerowano zbiór 5 cech istotnych ($q = 5$), które następnie zmodyfikowano, zastępując go $\binom{q}{2}$ nowymi cechami, zawierającymi w sobie informację o odległościach pomiędzy cechami przykładów.

Rozpatrzono wszystkie dwuelementowe kombinacje bez powtórzeń zbioru {1, 2, ..., 5}, odpowiadające im różnice wartości cech istotnych utworzyły następujące dziesięcioelementowe zbiory różnic:

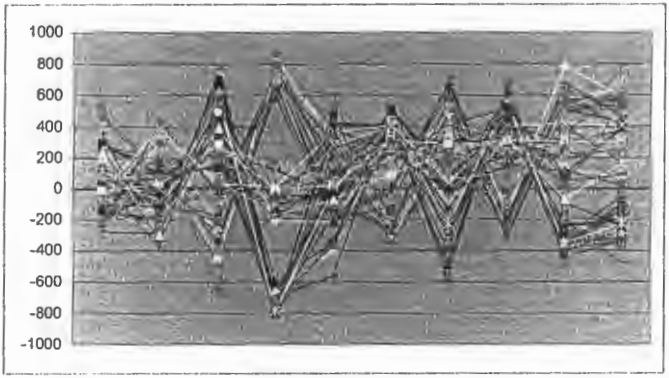
$$\{r_i(n)\}_{i=1}^{i=10} = \{\ y_2(n) - y_1(n), y_3(n) - y_2(n), y_4(n) - y_3(n), y_5(n) - y_4(n),$$

$$y_3(n) - y_1(n), y_4(n) - y_2(n), y_5(n) - y_3(n),$$

$$y_4(n) - y_1(n), y_5(n) - y_2(n),$$

$$y_5(n) - y_1(n)\ \}.$$

Na rys. 5 przedstawiono wartości cech $\{r_j(n)\}_{j=1}^{j=10}$ dla wszystkich szeregów czasowych ze zbioru uczącego, dla ustalonej permutacji cech.



Rysunek 5. Wartości cech dla wszystkich szeregów czasowych $\{r_j(n)\}_{j=1}^{j=10}$ ze zbioru uczącego, dla ustalonej permutacji cech

Cechy były następnie nominalizowane w sposób podany w Tabeli 1.

TABELA 1. NOMINALIZACJA RÓŻNIC WARTOŚCI CECH

Wartości	Wartości nominalne
< -800 and >= -1000	a
< -600 and >= -800	b
< -400 and >= -600	c
< -200 and >= -400	d
< -0 and >= -200	e
>= 0 and <= 200	f
> 200 and <= 400	g
> 400 and <= 600	h
> 600 and <= 800	i
> 800 and <= 1000	j

Przykładowo, wybrany szereg czasowy z klasy 1 można teraz zapisać za pomocą dziesięciu symboli w postaci: (c f e i f e h e i h).

Zbiory tak określonych przykładów (tzn. szeregów czasowych zapisanych w postaci symbolicznej), $\{r_j(n)\}_{j=1}^{j=10}$, stanowiły punkt wyjścia przy tworzeniu reguł.

Uwzględniając wszystkie 75 przykładów uczących z określoną dla nich przynależnością do jednej z trzech wygenerowanych klas, tworzono opisy tych klas stosując kilka wybranych metod uczenia maszynowego. Poniżej przedstawiono utworzone opisy klas:

Metoda G2:

klasa	r1	r2	r3	r4	r5	r6	r7	r8	r9	r10
1	*	*	*	h∨i∨j	*	*	*	*	*	*
2	*	*	*	a∨b∨c	*	*	*	*	*	*
3	*	*	*	d∨e∨f∨g	*	*	*	*	*	*

Uzyskano w 100% poprawną klasyfikację wszystkich szeregów ze zbioru uczącego.

Metoda IP1:

klasa	r1	r2	r3	r4	r5	r6	r7	r8	r9	r10
1	*	*	*	h∨i∨j	*	*	*	*	*	*
2	*	*	*	a∨b	*	*	*	*	*	*
	*	*	i	*	*	*	*	*	*	*
3	*	*	*	e∨f∨g	*	*	*	*	*	*
	*	*	*	*	*	*	*	*	f	*

Uzyskano w 100% poprawną klasyfikację wszystkich szeregów ze zbioru uczącego.

Dokładność klasyfikacji z zastosowaniem tak utworzonych opisów sprawdzana była dla zbioru testowego, zawierającego 75 nowych szeregów czasowych, które nie były stosowane w procesie uczenia sieci heteroasocjacyjnej. Zbiór testowy zawierał 25 szeregów z klasy 1, 25 szeregów z klasy 2 oraz 25 szeregów z klasy 3. Poniżej zamieszczono wyniki obliczeń.

Metoda G2:

Uzyskano w 100% poprawną klasyfikację wszystkich szeregów ze zbioru testowego.

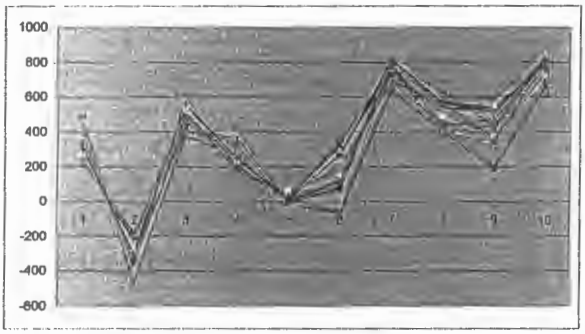
Metoda IP1:

Uzyskano w 98,7% poprawną klasyfikację szeregów czasowych ze zbioru testowego, jeden szereg czasowy (148) z klasy 3 został błędnie zaklasyfikowany do klasy 2.

4. Przykład obliczeniowy 3

(4-krokowe obwiednie górne)

Ilustrację graficzną wartości cech dla szeregów ze zbioru uczącego, dla ustalonej permutacji, z podziałem na klasy, zamieszczono na rys.4 a znominalizowane na rys.5.

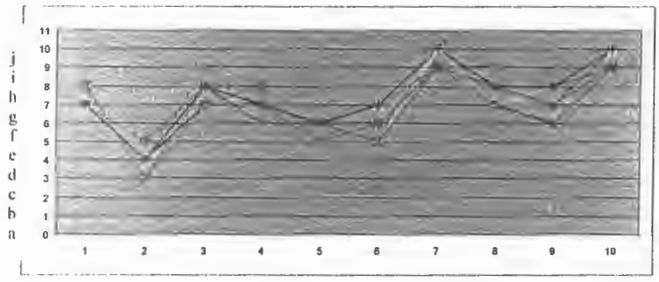


klasa 1 (ucz, max)

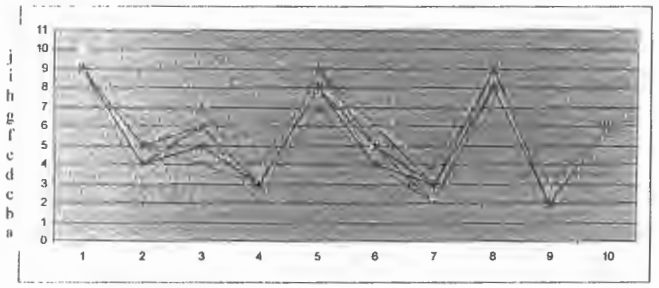
klasa 2 (ucz, max)

klasa 3 (ucz, max)

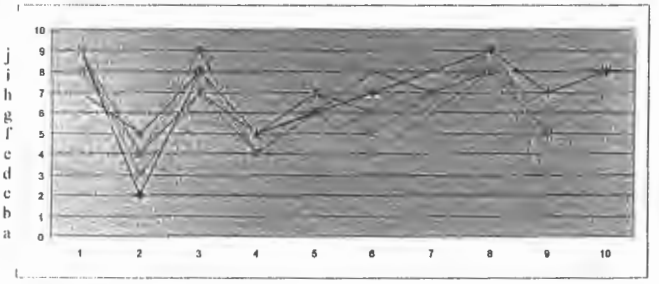
Rysunek 4. Wartości cech dla szeregów czasowych ze zbioru uczącego, dla ustalonej permutacji, z podziałem na klasy



klasa 1 (ucz, max)



klasa 2 (ucz, max)



klasa 3 (ucz, max)

Rysunek 5. Znominalizowane wartości cech szeregów ze zbioru uczącego, dla ustalonej permutacji, z podziałem na klasy

Poniżej przedstawiono wygenerowane opisy rozpatrywanych klas:

Metoda G1:

klasa	r1	r2	r3	r4	r5	r6	r7	r8	r9	r10
1	*	*	*	*	*	*	*	*	*	i∨j
2	*	*	*	*	*	*	*	*	b	*
	*	*	*	b∨c	*	*	*	*	*	*
3	*	*	*	*	*	*	*	*	*	g∨h
	*	*	*	*	*	*	f	*	*	*

Uzyskano w 100% poprawną klasyfikację wszystkich szeregów ze zbioru uczącego.

Metoda G2:

klasa	r1	r2	r3	r4	r5	r6	r7	r8	r9	r10
1	*	*	*	*	*	*	i∨j	*	*	*
2	*	*	*	*	*	*	b∨c∨d	*	*	*
3	*	*	*	*	*	*	e∨f∨g∨h	*	*	*

Uzyskano w 100% poprawną klasyfikację wszystkich szeregów ze zbioru uczącego.

Metoda IP1:

klasa	r1	r2	r3	r4	r5	r6	r7	r8	r9	r10
1	*	*	*	f∨g∨h	*	*		*	*	*
2	*	*	*	*	*	*		*	b	*
	*	*	*	c	*	*	*	*	*	*
	*	*	f	*	*	*	*	*	*	*
	*	f	*	*	*	*	*	*	*	*
3	*	*	*	*	*	*	e∨f∨g	*	*	*
	*	*	*	e	*	*	*	*	*	*

Uzyskano w 100% poprawną klasyfikację wszystkich szeregów ze zbioru uczącego.

Metoda IP2:

Poniżej przedstawiono wygenerowane opisy rozpatrywanych klas:

klasa	r1	r2	r3	r4	r5	r6	r7	r8	r9	r10
1	*	*	*		*	*	i∨j	*	*	*
2	*	*	*		*	*	b∨c	*	*	*
	*	*	*	c	*	*	*	*	*	*
3	*	*	*	*	*	*	e∨f∨g∨h	*	f	*

Uzyskano w 100% poprawną klasyfikację wszystkich szeregów ze zbioru uczącego.

Dokładność klasyfikacji z zastosowaniem tak utworzonych reguł sprawdzana była dla zbioru testowego, zawierającego 75 nowych szeregów czasowych, które nie były stosowane w procesie uczenia sieci heteroasocjacyjnej. Zbiór testowy zawierał 25 szeregów z klasy 1, 25 szeregów z klasy 2 oraz 25 szeregów z klasy 3.

Metoda G1:

Uzyskano w 97,3% poprawną klasyfikację szeregów czasowych ze zbioru testowego, jeden szereg czasowy (127) z klasy 2 został błędnie zaklasyfikowany do klasy 3 oraz jeden szereg (141) z klasy 3 do klasy 2.

Metoda G2:

Uzyskano w 98,7% poprawną klasyfikację szeregów czasowych ze zbioru testowego, jeden szereg czasowy (127) z klasy 2 został błędnie zaklasyfikowany do klasy 3.

Metoda IP1:

Uzyskano w 96% poprawną klasyfikację wszystkich szeregów czasowych ze zbioru testowego, jeden szereg czasowy (127) z klasy 2 został błędnie zaklasyfikowany do klasy 1, jeden szereg (136) z klasy 3 do klasy 1 oraz jeden szereg (141) z klasy 3 do klasy 2.

Metoda IP2:

Uzyskano w 98,7% poprawną klasyfikację wszystkich szeregów czasowych ze zbioru testowego, jeden szereg czasowy (127) z klasy 2 został błędnie zaklasyfikowany do klasy 3.

Literatura

1. Alcock, R. J., Manolopoulos, Y. (1999) Time-Series Similarity Queries Employing a Feature-Based Approach. 7th Hellenic Conference on Informatics, Ioannina, Greece.
2. Apostolico R., Bock M. E., Lonardi S. (2002) Monotony of surprise in large-scale quest for unusual words. In: Proceedings of the 6th International conference on research in computational molecular biology, Washington, DC, April 18-21, pp 22-31.
3. Benedikt, L., Kajic, V., Cosker, D., Marshall, D., Rosin, P. L. (2008) Facial Dynamics in Biometric Identification. In Proc. of British Machine Vision Conference, Leeds, 2008.
4. Gionis A., Mannila H. (2003) Finding recurrent sources in sequences. In: Proceedings of the 7th International conference on research in principles of database systems, Tucson, AZ, May 12-14, pp 249-256
5. Kacprzyk J., Szkatuła G. (1998) *IPI - An Improved Inductive Learning Procedure with a Preprocessing of Data.* Proceedings of IDEAL'98 (Hong Kong), Springer-Verlag.
6. Kacprzyk J., Szkatuła G. (1999) An inductive learning algorithm with a preanalysis od data. *International Journal of Knowledge - Based Intelligent Engineering Systems*, vol. 3, 135-146.
7. Kacprzyk J., Szkatuła G. (2002) An integer programming approach to inductive learning using genetic and greedy algorithms. W: L.C. Jain and J.Kacprzyk (eds.) *New learning paradigms in soft computing*. Studies in Fuzziness and Soft Computing. Physica-Verlag Heidelberg, 323 - 367.
8. Krawczak M., Miklewski A., Jakubowski A., Konieczny P. (2000) Investment Risk Management, (in Polish). Polish Academy of Sciences, Systems Research 25.
9. Krawczak M., Szkatuła G. (2010) On time series envelopes for classification problem. Developments of fuzzy sets, intuitionistic fuzzy sets, generalized nets, vol. II, 2010.
10. Krawczak M., Szkatuła G. (2010) Time series envelopes for classification. In: Proceedings of the conference: 2010 IEEE International Conference on Intelligent Systems, London, UK, July 7-9 2010, ss. 156-161.
11. Krawczak M., Szkatuła G. (2010) Redukcja wymiarowości szeregów czasowych. Studia i materiały Polskiego Stowarzyszenia Wiedzą, No. 31, 2010, ss. 32-45, 17 poz. bibl.
12. Kumar N., Lolla N., Keogh E., Lonardi S., Ratanamahatana C., Wei L. (2005) Time-Series Bitmaps: A Practical Visualization Tool for Working with Large Time Series Databases. In proceedings of SIAM International Conference on Data Mining (SDM '05), Newport Beach, CA, April 21-23, 2005.
13. Lin, J., Keogh, E., Lonardi, S., Chiu, B. (2003) A Symbolic Representation of Time Series, with Implications for Streaming Algorithms. Proceedings Data Mining and Knowledge Dicovering, San Diego.
14. Lin, J., Keogh, E., Wei, L., Lonardi, S. (2007) Experiencing SAX: a Novel Symbolic Representation of Time Series. Data Min Knowl Disc, 2, 15, 107–144.
15. Nanopoulos, A., Alcock, R., & Manolopoulos, Y. (2001): Feature-based Classification of Time-series Data. International Journal of Computer Research, 49-61.
16. Oja E. (1992) Principal components, minor components and linear neural networks. Neural Networks, vol.5, ss 927-935.
17. Pawlak Z. (1982): Rough Set. International Journal of Computer and Information Sciences, Vol. 11, No 5, 341-356.
18. Pawlak Z. (1991): Rough Set. Theoretical Aspect of Reasoning about Data. Kluwer Academic Publishers.
19. Roddick J. F., Hornsby K., Spilopoulos M. (2001): An updated bibliography of temporal, spatial and spatio-temporal data mining research. In Proceedings of the International Workshop on Temporal, Spatial and Spatio-Temporal data Mining. Berlin, Springer, Lecture Notes in Artificial Intelligence, 147-163.

20. Rodríguez, J.J. & Alonso, C.J. (2004): Interval and dynamic time warping-based decision trees. In Proceedings of the 2004 ACM symposium on Applied computing (SAC), 548-552.
21. Szkatula G. (1995) Machine learning from examples under errors in data, Ph.D. thesis, SRI PAS Warsaw, Poland.
22. Szkatula G. (2002). *Zastosowanie zmodyfikowanego zadania pokrycia w uczeniu maszynowym.* W: Gutenbaum J. (eds.): Automatyka Sterowanie Zarządzanie. SRI PAS, Warszawa, 431-445.
23. Wei, L., Keogh, E. (2006). Semi-Supervised Time Series Classification. In *Proc. of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2006),* pp. 748 - 753, Philadelphia, PA, U.S.A., August 20-23, 2006.
24. Wu, Y. & Chang, E.Y. (2004): Distance-function design and fusion for sequence data. CIKM '04, 324-333.
25. X. Xi, E. J. Keogh, C. R. Shelton, L. Wei, and C. A. Ratanamahatana. Fast time series classification using numerosity reduction. In ICML, 2006.