

Raport Badawczy

RB/52/2016

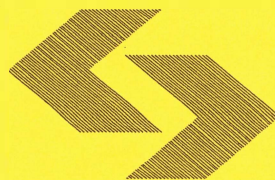
Research Report

**Uczenie komparatorów
obiektów złożonych**

Ł. Sosnowski

**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 Zakładu zgłaszający pracę:
Prof. dr hab. inż. Janusz Kacprzyk

Warszawa 2016

UCZENIE KOMPARATORÓW OBIEKTÓW ZŁOŻONYCH

Łukasz Sosnowski

Instytut Badań Systemowych Polskiej Akademii Nauk
e-mail: Lukasz.Sosnowski@ibspan.waw.pl

Streszczenie. Artykuł przedstawia zagadnienie uczenia zbioru referencyjnego dla zadanego komparatora obiektów złożonych. Metoda jest rozszerzalna na całą sieć komparatorów. Artykuł zawiera również kompendium wiedzy o teorii komparatorów obiektów złożonych. Uczenie zbioru jest metodą bazująca na samoorganizacji i dzięki jej wprowadzeniu teoria komparatorów w pełni zalicza się do metod sztucznej inteligencji, używanych do rozpoznawania i identyfikacji obiektów złożonych.

Słowa kluczowe: komparator obiektów złożonych, podobieństwo, rozpoznawanie i identyfikacja obiektów, uczenie maszynowe, samoorganizacja zbioru obiektów

1 WPROWADZENIE

Sztuczna inteligencja zajmuje się między innymi wieloma aspektami rozpoznawania wzorców. Jedną z jej dziedzin są systemy bazujące na obserwacji podobieństwa pomiędzy obiektami i wnioskowanie na tej podstawie istnienia pewnych relacji powiązanych z obiektami wejściowymi (podlegającymi rozpoznawaniu). Komparatory obiektów złożonych stanowią zunifikowane podejście do budowy systemów rozpoznawania obiektów złożonych bazując na wiedzy o podobieństwie obiektów. Aby w pełni można zaliczyć wybrane podejście do klasy metod inteligentnych, należy rozważyć możliwości uczenia występujące dla danej metody.

W systemach decyzyjnych można wyróżnić wiele metod uczenia i pozyskiwania wiedzy. Ich podział opiera się na wielu kryteriach. Głównym kryterium jest strategia uczenia oraz wyjście uzyskiwane z algorytmu [1]. Wśród algorytmów uczących można wyróżnić więc następujące typy:

- Uczenie z nadzorem - realizowane poprzez stworzenie funkcji mapującej wejścia oraz zadane wyjścia. Typowym przykładem jest problem klasyfikacji.
- Uczenie bez nadzoru - realizowane jedynie w oparciu o dane wejściowe przy użyciu metod wykrywających pewne regularności lub podobieństwa obiektów.
- Uczenie z częściowym nadzorem - połączenie obu powyższych metod w celu wygenerowania funkcji klasyfikującej.
- Uczenie ze wzmocnieniem ma na celu automatyczne pozyskiwanie wiedzy proceduralnej w oparciu o interakcję ze środowiskiem. Każde działanie powoduje pewną reakcję środowiska, która jest wskazówką dla algorytmu.
- Transdukcja - podejście podobne do uczenia z nadzorem, cho wynikiem uczenia nie jest funkcja, tylko predykcja możliwych wartości wyjściowych, które bazują na danych wejściowych treningowych.
- Meta-uczenie - algorytm wykorzystujący różne inne metody uczenia do weryfikacji zbioru hipotez. Jej celem jest selekcja najlepszej metody uczenia dla zadanego problemu.

W niniejszym artykule opisane zostały metody uczenia dotyczące pojedynczych komparatorów, czyli uczenie w kontekście poszczególnych cech obiektu rozpoznawanego. Artykuł składa się z 5 sekcji. Pierwsza zawiera informacje wprowadzające, nakreślające kontekst artykułu. Druga sekcja przedstawia usystematyzowaną teorię komparatorów obiektów złożonych, stanowiącą główny przedmiot metodyki rozpoznawania obiektów złożonych przedstawionej w poprzednich publikacjach [14, 12, 16]. Sekcja trzecia omawia w skrócie wariant uczenia bez nadzoru, charakterystykę tego podejścia. Rozdział czwarty stanowi opis właściwego algorytmu uczenia w odniesieniu do komparatorów obiektów złożonych, z uwzględnieniem dwóch kluczowych aspektów: uczenia zbioru referencyjnego oraz uczenia parametru progowania [13]. Ostatnia sekcja stanowi podsumowanie oraz omówienie planowanych przyszłych prac związanych z tematyką uczenia komparatorów.

2 KOMPARATORY OBIEKTÓW ZŁOŻONYCH

Dotychczas komparator znany był jako wyspecjalizowany element logiczny służący do wykonywania porównań pomiędzy dwiema liczbami otrzymywanymi na wejściu. Określał wartość relacji większości pomiędzy nimi. Najczęściej spotykane były komparatory binarne służące

do porównywania liczb zakodowanych w systemie dwójkowym. Pojęcie komparatora zyskało na popularności w latach pięćdziesiątych XX wieku i szybko znalazło zastosowanie w elektronice. Stosowane do dziś w postaci komparatora analogowego oraz komparatora cyfrowego, specjalizują się w porównywaniu sygnałów analogowych i cyfrowych. Komparatory tego rodzaju stanowią atomowe elementy wchodzące w skład większych podzespołów.

Dużym krokiem w ewolucji komparatorów był rozwój języków programowania. Zaczęło powstawać coraz więcej programów komputerowych. Programy te operowały na danych, które przechowywane były w strukturach i obiektach. Podstawową operacją wykonywaną podczas obliczeń była operacja porównania. Dawała ona możliwość stwierdzenia, w jakiej wzajemnej relacji znajdują się elementy wejściowe. W tym przypadku na wejściu komparatory zaczęły otrzymywać już nie tylko liczby, ale i proste typy danych odpowiadające atrybutom obiektu. Odpowiedź komparatora pozostawała jednak niezmienna w postaci pojedynczej wartości określającej stan relacji większości.

Komparator obiektów złożonych jest analogicznym bytem, przystosowanym jednak do przetwarzania bardziej kompleksowych danych w postaci całych obiektów. Oznaczany jest jako com^{ref} , natomiast można go interpretować jako funkcję w postaci:

$$\mu_{com}^{ref} : X \times 2^{ref} \rightarrow [0, 1]^{ref}, \quad (1)$$

gdzie $X \subseteq U$ jest zbiorem obiektów wejściowych, czyli tych, które będą porównywane, ref to zbiór obiektów, względem których podobieństwo będzie badane. Dodatkowo $a(x)$ jest funkcją generującą reprezentację obiektu $x \in X$ do przetwarzania przez dany komparator dla zadanej cechy a . Analogiczna funkcja reprezentacji używana jest dla poszczególnych obiektów referencyjnych $a(y)$, gdzie $y \in Y$. W dalszej części publikacji zbiór ref będzie nazywany zbiorem referencyjnym, a Y podzbiorem referencyjnym ($Y \subseteq ref$). Przy ustalonym porządku na zbiorze $ref = \{y_1, \dots, y_{|ref}|\}$ komparator może być zapisany jako funkcja w postaci

$$\mu_{com}^{ref}(x, Y) = Sh(F(\mathbf{v})), \quad (2)$$

gdzie Sh jest funkcją wyostrzania wyniku, F jest funkcją odpowiedzialną za filtrowanie wyników częściowych, opartą o funkcje idempotentne typu: min , max , top , itp., a \mathbf{v} jest wektorem bliskości pomiędzy obiektem wejściowym x a wszystkimi elementami zbioru ref . W przypadku, gdy Y nie jest podzbiorem właściwym ref , na współrzędnych

wektora \mathbf{v} odpowiadających elementom $y_i \notin Y$ znajdować się będą wartości zerowe. Wartościami pozostałych współrzędnych tego wektora są podobieństwa poszczególnych par obiektów, powstałych z połączenia obiektu wejściowego x oraz elementów podzbioru referencyjnego Y . Wymiarem wektora \mathbf{v} jest moc zbioru referencyjnego ref . Uwzględniając powyższe, wektor bliskości przyjmie postać:

$$\mathbf{v}[i] = \begin{cases} 0 & : y_i \notin Y \\ sim(x, y_i) & : y_i \in Y \end{cases} \quad (3)$$

Wartości podobieństw (ozn. $sim(u, y)$) obliczane są za pomocą relacji rozmytej [4], złożonej z dodatkowymi mechanizmami kontrolującymi powstałe wyniki.

Budowa komparatora obiektów złożonych, sposób jego działania oraz wejście i wyjście stanowią istotną różnicę względem wcześniejszych podejść. Poprzednie podejścia bazowały na atomowych wartościach przekazywanych do komparatora. Wynik również by prostą ustaloną odpowiedzią, np. $\{0, 1\}$ lub $\{1, -1, 0\}$, gdzie poszczególne elementy wyrażały wartość relacji pomiędzy wejściowymi elementami (większości, mniejszości lub równości) [5]. W przypadku komparatorów obiektów złożonych badane jest podobieństwo, a nie wartość pojedynczej relacji. To jedna z podstawowych różnic pomiędzy tymi podejściami, która jest jednocześnie diametralna. Fakt ten umożliwia analizę danego problemu w znacznie szerszym kontekście. Przedstawione w niniejszej publikacji podejście zawiera kilka autorskich rozwiązań niespotykanych dotąd w dziedzinie komparatorów. Jednym z nich jest wspomniany wcześniej zbiór referencyjny. Zbiór ten zawiera instancje obiektów, które reprezentują wiedzę dziedzinową dotyczącą badanego problemu. Stanowi ona swoistą składnicę wiedzy, która jest wykorzystywana do porównań i konstrukcji wektora bliskości. Stanowi ona zbiór elementów, za pomocą których wyrażane jest dane podobieństwo. Zagadnienia konstrukcji tego rodzaju zbioru zostały opisane w publikacji [19].

Innym nowatorskim rozwiązaniem dotyczącym komparatorów są reguły wzbraniające [14]. Umożliwiają one wyłączenie części rozwiązań z przestrzeni rozważań, bez zmiany podstawowej funkcji przynależności do relacji (funkcji podobieństwa bazowego). Reguły te stanowią implementację mechanizmu wyjątków od zdefiniowanych funkcji wyliczania podobieństwa. Upraszczają w ten sposób konstrukcję komparatorów i zwiększają ich czytelność oraz łatwość zarządzania zmianami.

Kolejnym nowym rozwiązaniem wprowadzonym do komparatorów jest mechanizm kontroli jakości rozwiązań [15]. Polega on na tym, iż rozwiązania o zbyt niskiej wartości podobieństwa nie są brane pod uwagę. Wartość progu, dla którego rozwiązanie przestaje być interesujące, ustalana jest indywidualnie. Jest jednym z parametrów komparatora, który może podlegać uczeniu.

Komparatory tego rodzaju charakteryzują się jeszcze dwiema istotnymi różnicami. Pierwszą jest mechanizm separujący wyniki, zwany mechanizmem wyostrzania. Jego zadaniem jest wzmocnienie różnic pomiędzy najlepszymi rozwiązaniami a średnimi i słabymi. Druga wspomniana różnica to występowanie bloku filtracji rozwiązań, który określa przydatność danego rozwiązania na podstawie porównania z wynikami innych par porównywanych obiektów. Z tego punktu widzenia jest to mechanizm oparty na konkurencyjności pomiędzy obiektami referencyjnymi, a nie na pewnej niezależnej regule, tak jak przy wspomnianym wcześniej mechanizmie kontrolowania jakości poszczególnych rozwiązań. Współzawodnictwo wyrażane jest poprzez porównanie wyników pomiaru podobieństwa poszczególnych par obiektów.

Niezaprzeczalną zaletą komparatorów jest ich uniwersalność, która wiąże się z metodą postępowania względem dowolnego typu przetwarzanego obiektu. Technika ta może być użyta do rozwiązywania problemów decyzyjnych związanych z obiektami tekstowymi, obiektami graficznymi, materiałami wideo lub audio itp. [13]. Zastosowane wewnątrz metody pomiaru podobieństwa oraz zbiory referencyjne są różne, dobierane indywidualnie w zależności od danego problemu. Jednakże podstawowa zasada działania pozostaje niezmienna. W dalszej części przedstawiono poszczególne elementy składowe komparatora obiektów złożonych.

2.1 ZBIÓR REFERENCYJNY

Niepusty zbiór uporządkowany $ref = \{y_1, \dots, y_{|ref|}\}$, który stanowi podzbiór obiektów uniewersum U , wybrany ze względu na obiekty wejściowe $x \in X$, to zbiór referencyjny. Zbiór ten składa się z obiektów, względem których wykonywane są porównania w komparatorze. Obiekty te są zgrupowane w jeden zbiór ze względu na pewne łączące je relacje i atrybuty lub pewną reprezentowaną wiedzę. Poszczególne elementy są odrębnymi instancjami obiektów z kombinacjami wartości atrybutów, rozpinającymi przestrzeń możliwych stanów cech obiektów, które są istotne dla zadanego komparatora lub grupy komparato-

rów. Zbiór referencyjny stanowi pewnego rodzaju analogię do pojęcia decyzji uogólnionych, które występują w zbiorach przybliżonych [7]. Pojęcie to określa zbiór klas decyzyjnych, które są możliwe dla danego obiektu. Jednakże komparatorów nie dotyczą klasy decyzyjne, tylko podzbiory obiektów referencyjnych możliwe dla danego obiektu wejściowego, co jest do pewnego stopnia analogiczne do wspomnianego pojęcia.

Zbiór referencyjny jest dla komparatora niezbędny dla celów porównań i zwracania wartości podobieństwa. Jednakże dla konkretnego komparatora na wejściu przekazywany jest podzbiór referencyjny $Y \subseteq ref$, którego elementy w konkretnym komparatorze używane są do porównań.

2.2 PODOBIENSTWO BAZOWE

Funkcja przynależności do relacji rozmytej [4], zdefiniowana jako:

$$\mu : X \times ref \rightarrow [0, 1], \quad (4)$$

gdzie ref jest zbiorem referencyjnym, a X zbiorem obiektów wejściowych. Zgodnie z definicją funkcja ta osiąga wartości z przedziału $[0, 1]$, gdzie 0 oznacza brak podobieństwa, a 1 całkowitą przynależność do relacji, co w przypadku komparatorów interpretowane jest odpowiednio jako brak podobieństwa lub nierozróżnialność względem badanej cechy. W danym komparatorze zdefiniowana jest dokładnie jedna funkcja przynależności. Wykorzystywana jest do obliczania wartości poszczególnych współrzędnych wektora bliskości (wzór (3)). Różne komparatory mogą posiadać różne funkcje przynależności, aczkolwiek nie jest to wymagane. Istnieje zatem możliwość użycia tej samej funkcji podobieństw bazowych w wielu komparatorach, lecz jej wybór podyktowany jest głównie rodzajem przetwarzanych obiektów oraz rozwiązywanym problemem.

2.3 FUNKCJA PROGOWA

Funkcja zdefiniowana jako:

$$t_h(z) = \begin{cases} 0 & z < p \\ z & z \geq p \end{cases}, p \in [0, 1], \quad (5)$$

gdzie z jest argumentem funkcji reprezentującym wartość funkcji podobieństwa bazowego dla danej pary obiektów (x, y) w danym komparatorze [23]. Głównym zadaniem tej funkcji jest ograniczenie zbyt słabych rozwiązań, np. takich, dla których podobieństwo jest zbyt niskie.

Jeśli wartość parametru p nie zostanie osiągnięta, wartość podobieństwa ustalana jest na 0. Parametr p jest określany indywidualnie dla każdego komparatora. Może być ustalony przez eksperta lub podlegać uczeniu maszynowemu.

Mechanizm ten związany jest z pojęciem α -przekroju zbioru rozmytego [4]. Każdy komparator obiektów złożonych konstruuje zbiór rozmyty. Funkcja progowania przekształca powstały zbiór w jego p -przekrój poprzez zastosowanie funkcji danej wzorem (5).

2.4 REGUŁY WZBRANIAJĄCE

Zdefiniujmy rodzinę zbiorów indeksowanych zmienną i taką, że dla każdego obiektu referencyjnego y_i istnieje zbiór reguł $Rules_i$. Zbiór ten zawiera reguły dotyczące obiektu referencyjnego y_i , lecz ich argumentem jest obiekt wejściowy $x \in X$. Pojedyncza reguła może być zapisana w postaci predykatów powiązanych operatorami logicznymi, takimi jak koniunkcja, alternatywa czy negacja [3]. Jeśli dane zdanie logiczne jest prawdziwe, to reguła jest spełniona. Ogólna postać reguły to:

$$r_j : X \rightarrow \{0, 1\}, \quad (6)$$

gdzie j jest indeksem reguł w zbiorze $Rules_i$, który oznacza numer zdefiniowanej reguły. Cały mechanizm może być modelowany w postaci funkcji:

$$Exc_{Rules_i}^{ref}(x) = \max_{j=1}^{|Rules_i|} \{r_j(x)\}, \quad x \in X \quad (7)$$

Od strony funkcjonalnej reguły te umożliwiają wyłączenie części rozwiązań z przestrzeni rozważań, bez zmiany podstawowej funkcji podobieństwa bazowego. Upraszczają w ten sposób konstrukcję komparatorów i zwiększają ich czytelność oraz łatwość zarządzania zmianami.

2.5 FUNKCJA PODOBIEŃSTWA

Funkcja określająca podobieństwo pomiędzy pojedynczą parą obiektów (x, y) , spełniająca:

$$sim : X \times ref \rightarrow [0, 1] \quad (8)$$

Określa ona wartość poszczególnych współrzędnych w wektorze bliskości v (wzór (3)). Stanowi złożenie trzech zdefiniowanych wcześniej elementów: funkcji podobieństwa bazowego, funkcji progowania oraz

reguł wzbraniających. Ogólna postać tej funkcji wyrażana jest wzorem:

$$sim(x, y_i) = \begin{cases} 0 : & Exc_{Rules_i}^{ref}(x) = 1 \vee y_i \notin Y \\ t_h(\mu(x, y_i)) : & w.p.p. \end{cases}, \quad (9)$$

gdzie $x \in X, y_i \in ref$, a t_h oznacza funkcję progową, μ jest funkcją podobieństwa bazowego, $Exc_{Rules_i}^{ref}$ jest funkcją reguł wzbraniających oraz i jest indeksem współrzędnej wektora bliskości, dla której obliczane jest podobieństwo, a $Y \subseteq ref$.

2.6 FUNKCJA FILTRUJĄCA

Funkcja oznaczona jako F filtrująca wartości wektora v . Dobór funkcji znacząco wpływa na właściwości komparatora. Wynikiem działania funkcji jest zawsze wektor w postaci:

$$F(v) = \langle h(v[1]), \dots, h(v[|ref|]) \rangle, \quad (10)$$

gdzie $v[i] = h(v[i])$ oraz h jest jedną z funkcji idempotentnych. Przykładem prostej funkcji filtrującej jest funkcja max , która wybiera wyniki o najwyższej wartości.

Istotą funkcji filtrującej jest dokonanie filtracji wyników w taki sposób, aby zminimalizować liczbę wyników niezerowych zwracanych przez komparator. W pewnym sensie jest to funkcja przekształcająca wektor bliskości w wektor rzadki (choć nie w każdym przypadku). Filtracja odbywa się przy współdziałaniu wartości wszystkich współrzędnych. Realizuje założenie współzawodnictwa poszczególnych obiektów referencyjnych między sobą. Mechanizm ten operuje na pełnym zbiorze wyników wytworzonym na podstawie wyliczenia podobieństwa dla każdej pary obiektu wejściowego z poszczególnymi obiektami referencyjnymi. Dzięki temu filtracja uwzględnia wszystkie kombinacje wyników oraz jest w stanie je porównać. Opiera się zatem na wyborze najlepszych wyników niezależnie od poziomu pojedynczych wartości podobieństwa (kontrolowanych przez inny, wcześniej opisywany mechanizm).

2.7 FUNKCJA WYOSTRZANIA

Funkcja mająca na celu zwiększenie odległości pomiędzy wynikami najlepszymi, średnimi oraz najslabszymi. Jej działanie koncentruje się

na analizie poszczególnych wartości wektora bliskości. Dziedzina oraz przeciwdziedzina tej funkcji jest wektor o wymiarze $|ref|$. Funkcja ta ma posta:

$$Sh : [0, 1]^{ref} \rightarrow [0, 1]^{ref}, \quad (11)$$

Wyostżranie może odbywać się za pomocą różnych przekształceń, jednakże w niniejszej pracy używane będzie następujące:

$$Sh(\mathbf{v})[i] = \begin{cases} max_v \cdot e^{\alpha(v[i]-max_v)}, & \alpha > 0, \\ 0, & w p.p. \end{cases} \quad (12)$$

gdzie \mathbf{v} jest wektorem bliskości, a max_v jest maksymalną wartością współrzędnej wektora \mathbf{v} w postaci:

$$max_v = max_{i=1}^{|ref|} \{v[i]\} \quad (13)$$

Funkcja (12) jest nieliniową funkcją o ciekawej własności dla współczynnika $\alpha > 0$ [20]. Charakteryzuje się trzema podstawowymi cechami:

$$\forall i \in \{1, \dots, |ref|\} : (v[i] = 0) \Rightarrow (Sh(\mathbf{v})[i] = 0), \quad (14)$$

gdzie wzór (14) oznacza zachowanie niezmiennych wartości równych zero, dzięki czemu wynik nie zostaje sztucznie zawyżony;

$$\forall i \in \{1, \dots, |ref|\} : (v[i] = max_{j=1}^{|ref|}(v[j])) \Rightarrow (Sh(\mathbf{v})[i] = v[i]), \quad (15)$$

gdzie wzór (15) oznacza zachowanie niezmiennych wartości maksymalnych, dzięki czemu najlepszy wynik zachowuje swoje pierwotne własności;

$$\forall i, j \in \{1, \dots, |ref|\} : (v[i] < v[j]) \Rightarrow (Sh(\mathbf{v})[i] < Sh(\mathbf{v})[j]), \quad (16)$$

gdzie wzór (16) oznacza silną monotoniczność, co jest niezbędnym warunkiem do spełnienia zakładanej roli.

3 UCZENIE BEZ NADZORU

Metoda stosowana w odniesieniu do problemów, dla których zbiory testowe albo nie istnieją, albo bardzo trudno jest je skonstruować. Metoda ta polega na wykorzystaniu wiedzy wynikającej bezpośrednio z obiektów (z danych), jej wykryciu, analizie i modelowaniu. Sprowadza

się do analizy skupień w danych na podstawie różnego rodzaju kryteriów, m.in. kryterium podobieństwa obiektów. Grupowanie danych to realizacja uczenia bez nadzoru.

Poprzez odkrywanie wiedzy z danych identyfikowane są cechy, względem których dokonywane jest grupowanie. Problem automatycznego grupowania jest zagadnieniem o wysokim stopniu skomplikowania. Nie istnieje jeden uniwersalny algorytm grupowania dla każdego rodzaju danych. W każdym przypadku należy użyć wiedzy dziedzinowej o problemie i dobrać algorytm indywidualnie. Dodatkowym utrudnieniem jest to, iż w większości przypadków nie jest znana liczba grup, która ma powstać w wyniku działania algorytmu. Istnieje jednak pewien warunek, który grupy danych powinny spełniać. Składa się on z dwóch punktów:

1. Homogeniczność wewnątrz grupy
2. Heterogeniczność pomiędzy grupami

Punkt pierwszy oznacza, iż w ramach danej grupy elementy powinny być jak najbardziej do siebie podobne, natomiast punkt drugi określa, iż elementy różnych grup powinny być jak najbardziej od siebie odmienne.

Procedura uczenia bazuje na dostępnych danych, lecz bez nadanej etykiety decyzyjnej, przetwarzając je jedna po drugiej. Struktura grup (ang. clusters) zmienia się w procesie przetwarzania. Konstrukcja klastrów w trakcie przetwarzania zależy od kolejności przetwarzania obiektów z danych, lecz po przetworzeniu całego zbioru powinno się uzyskać ten sam wynik. Istotnym elementem są również parametry dotyczące ewaluacji wyniku, akceptowalnego podobieństwa itd. W zależności od tych parametrów powstaje na wyjściu różna liczba grup obiektów. Liczba ta jest istotnym czynnikiem, który wpływa na jakość grupowania. Jako kryterium oceny poprawności grupowania stosuje się tzw. wskaźniki jakości grupowania [10]. Wskaźników tych jest wiele. Podstawowy z nich bazuje na ocenie odległości wszystkich obiektów od środków grup.

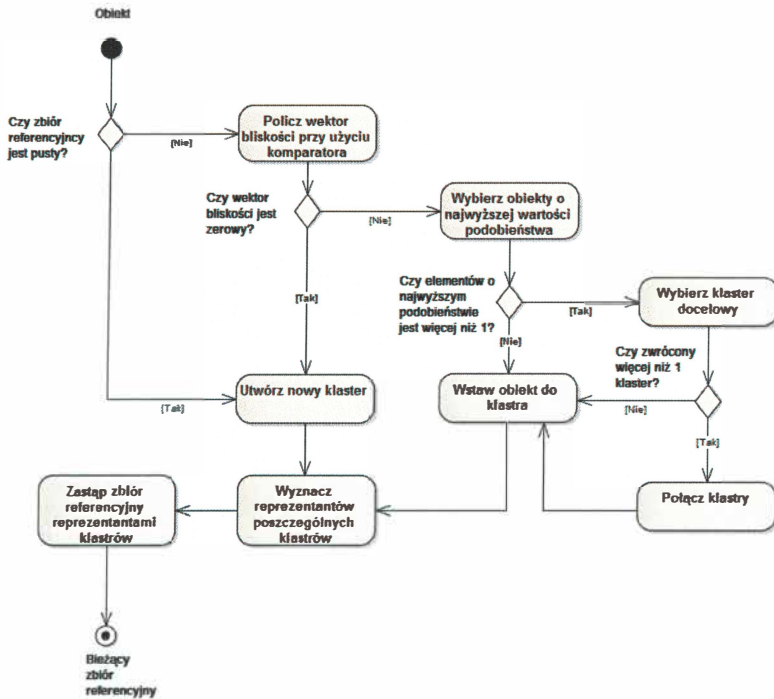
4 ZAGADNIENIE UCZENIA POJEDYNCZEGO KOMPARATORA

Jednym z zagadnień pojawiającym się przy tworzeniu komparatora oraz budowaniu rozwiązania opartego na tej metodyce jest odpowiednie skonstruowanie zbioru referencyjnego. Zbiór referencyjny zawie-

ra wybrane obiekty. W praktyce przetwarzane są instancje obiektów, gdyż są to konkretne egzemplarze danego typu obiektu. Jednakże na potrzeby niniejszej publikacji terminy instancja obiektu oraz obiekt będą stosowane wymiennie. Opisywany problem sprowadza się do odpowiedniego dobrania obiektów charakteryzujących się takimi cechami i wartościami cech, dzięki którym zbiór obiektów będzie reprezentatywny dla problemu postawionego do rozwiązania, przy uwzględnieniu właściwości komparatora (badana cecha), jak również specyfiki obiektu wejściowego. Uczenie zbioru referencyjnego będzie zatem automatyczną oraz inteligentną metodą wyznaczenia zbioru, który będzie w stanie reprezentować większość przetwarzanych obiektów.

Wyznaczenie zbioru referencyjnego może być przeprowadzone w przypadku posiadania zbioru instancji obiektów oraz ustalonego komparatora. Metoda ta bazuje na klastrowaniu zbioru instancji przy użyciu funkcji podobieństwa. Do przeprowadzenia procedury uczącej wykorzystywana jest metoda walidacji krzyżowej w wersji leave-one-out. Ten rodzaj walidacji zakłada, iż dany zbiór wejściowy zostaje podzielony n -krotnie na dwa podzbiory, przy czym pierwszy zbiór jest jednoelementowy. Proces uczenia polega na samoorganizacji obiektów w klastry. Metoda wyznaczania klastra bazuje na podobieństwie obliczanym przez komparator. Z poszczególnych klastrów wybierany jest obiekt reprezentujący dany klaster w zbiorze referencyjnym. Istnieje kilka znanych metod wyboru reprezentantów klastrów. Jedne oparte są o punkty ciężkości (centroidy), inne o maksymalizację lub minimalizację odległości pomiędzy poszczególnymi klastrami. W przypadku komparatorów stosowane mogą być dowolne z tych metod.

Pojedyncza iteracja uczenia została przedstawiona na rys. 1. Algorytm startuje z pustym zbiorem referencyjnym. W takim przypadku pierwszy obiekt wejściowy staje się automatycznie jednoelementowym klastrzem oraz reprezentantem w zbiorze referencyjnym. W każdej iteracji algorytmu wyznaczany jest wektor bliskości względem aktualnych reprezentantów klastrów. Jeśli komparator zwróci wektor niezerowy, obiekt wejściowy przypisywany jest do klastra reprezentowanego przez obiekt referencyjny, dla którego uzyskano największe podobieństwo. W przypadku, gdy więcej niż jeden obiekt referencyjny uzyskał taką samą najwyższą wartość podobieństwa, należy zastosować jedną z metod wyboru klastra, opisaną w literaturze [2]. Wybrana metoda może wskazać równorzędnie więcej niż jeden klaster, co powinno



Rys. 1. Diagram aktywności w notacji UML wykonania jednej iteracji w uczeniu bez nadzoru obiektów referencyjnych

skutkować połączeniem klastrów w jeden, gdyż obiekt zasila dokładnie jeden klastor.

W przypadku, gdy wektor bliskości zwrócony został jako zerowy, obiekt ten tworzy nowy klastor, automatycznie go zasilając. Za wielkość klastra odpowiadają parametry p komparatorów, określające minimalne akceptowalne podobieństwo na wyjściu komparatora. Dlatego też w przypadku chęci uzyskania mniejszej liczby klastrów, przy automatycznym zwiększeniu tolerancji dla różnic pomiędzy obiektami w ramach jednego klastra, należy obniżyć wartości parametrów p .

Analogiczna procedura stosowana jest w przypadku każdego kolejnego elementu pochodzącego ze zbioru walidacji krzyżowej, aż do wyczerpania obiektów. Należy zwrócić uwagę na fakt, iż ustawienie zbyt rygorystycznych kryteriów jakości podobieństwa może spowodo-

wać utworzenie oddzielnych klastrów dla każdego obiektu. Natomiast w przypadku, gdy wartości parametrów ustawione zostaną zbyt nisko, może się zdarzyć, iż wszystkie obiekty znajdują się w jednym klastrze.

Innym przykładem uczenia komparatora obiektów złożonych jest optymalizacja parametru progowania p . Zgodnie z definicją przytoczoną w sekcji 2.3, parametr p odpowiada za minimalną dopuszczalną wartość podobieństwa występującą w wektorze bliskości danego komparatora. W celu jego automatycznego wyznaczenia należy posłużyć się metodą nadzorowaną, tzn. taką, gdzie do dyspozycji będzie pewien zbiór danych, na podstawie których można będzie nauczyć algorytm żądanej wartości. Do wykonania tej procedury można użyć walidacji krzyżowej w trybie wielokrotnego próbkowania (min. 10-krotnego). Zbiór danych dzielimy na dwa podzbiory. Pierwszy nazywany jest zbiorem uczącym o licznosci $\frac{1}{3}$ wszystkich elementów, za drugi - zbiorem testowym o licznosci $\frac{2}{3}$ wszystkich elementów. W przypadku pojedynczego komparatora zagadnienie uczenia jest dość proste. Należy wyznaczyć wartość graniczną podobieństwa, poniżej której nie znajduje się żadne prawidłowe rozwiązanie. W przypadku komparatorów są to podobieństwa przypisane do obiektów referencyjnych, które stanowią odpowiedź komparatora. Jednocześnie przy identyfikacji są to obiekty z maksymalną wartością podobieństwa. Do wyznaczenia tej wartości można użyć różnych metod. Jedną z nich jest proste przeszukiwanie lokalne [22]. Punktem startowym algorytmu będzie wartość 1. Jakość rozwiązania będzie określona przez funkcję w postaci:

$$f_{eval}(p) = \sum_{u \in U} (f_{recall}(x) - (1 - p)), \quad (17)$$

gdzie

$$f_{recall} : X \rightarrow \{0, 1\} \quad (18)$$

jest funkcją zwracającą wartość 1 przy uzyskaniu poprawnej etykiety decyzyjnej zbioru treningowego dla pary obiektów (x, y) lub w przeciwnym przypadku 0.

Im wyższa wartość funkcji (17), tym wyższa jakość rozwiązania. Rozwiązania sąsiednie generowane są poprzez modyfikację wartości parametru o ustaloną stałą, np. 0.01 (dodanie lub odjęcie). Warunek stopu może być zaimplementowany w postaci monitorowania powtarzających się rozwiązań. Po przekroczeniu wartości granicznej jakości rozwiązania zacznie spadać, dlatego wartość zacznie wahadłowo rosnąć, a następnie maleć. W momencie wykrycia tego zjawiska należy przerwać przetwarzanie.

5 PODSUMOWANIE

Przedstawione algorytmy uczenia pozwalają na dobranie niezbędnych parametrów i danych dla komparatora obiektów złożonych w celu optymalizacji jego działania. Dzięki temu każdy pojedynczy komparator może zostać uczeniu, co w konsekwencji może spowodować optymalizację szerszego rozwiązania, np. w postaci całej sieci komparatorów. Przedstawione podejście pozwala m. in. na automatyczne rozwiązanie fundamentalnego problemu dla metod bazujących na komparatorach jakim jest konstrukcja zbioru referencyjnego.

W niniejszej publikacji znajduje się jedynie opis teoretyczny rozwiązania. Zadaniem do rozwiązania w dalszych badaniach pozostaje przeprowadzenie eksperymentów wykazujących w jaki sposób zmienia się charakterystyka i efektywność metody rozpoznawania obiektów w zależności od wyuczonych parametrów i skonstruowanego zbioru referencyjnego.

Jednakże dzięki opracowaniu opisanych metod uczenia, komparatory obiektów złożonych zyskały silne narzędzie umożliwiające adaptację skonstruowanego rozwiązania do warunków napotkanych w rozpatrywanych danych. Metoda wraz z opracowanymi algorytmami uczenia (uczenie struktury) może zostać zaadoptowana do iteracyjnego uczenia zbioru referencyjnego przy jednoczesnej optymalizacji struktury, co pośrednio odpowiada poszukiwaniu bireduktów, znanych ze zbiorów przybliżonych i eksploracji danych [21].

Literatura

1. T.O. Ayodele, Introduction to Machine Learning, INTECH Open Access Publisher, 2010.
2. M. Brun, C. Sima J. Hua, J. Lowey, B. Carroll, E. Suh, E. R. Dougherty, Model-based evaluation of clustering validation measures, Pattern Recognition, vol. 40, no. 3, pp. 807 - 824, 2007.
3. M. Gruber, Mastering SQL, SYBEX Inc., Alameda, CA, USA, 2000
4. J. Kacprzyk. Multistage Fuzzy Control: A Model-based Approach to Fuzzy Control and Decision Making. John Wiley Sons, Limited, 2012.
5. M. Nesenbergs and V. O. Mowery, Logic Synthesis of Some High-Speed Digital Comparators, Bell System Technical Journal, vol. 38, 1959.
6. D. Nohlen, Prawo wyborcze i system partyjny: o teorii systemów wyborczych. Scholar, 2004. [Online]. Available: <http://books.google.pl/books?id=ckODAAAACAAJ>
7. Z. Pawlak, Rough Set Theory, KI, vol 15, no. 3, pp. 38-39, 2001.
8. L. Polkowski, Approximate Reasoning by Parts: An Introduction to Rough Mereology, ser. Intelligent Systems Reference Library. Springer, 2011.
9. D. E. Rumelhart, G. E. Hinton, R.J. Williams, Learning Representations by Back-propagating Errors, pp. 696-699, MIT Press, Cambridge, MA, USA, 1988.

10. L. Rutkowski, *Computational Intelligence: Methods and Techniques*, Springer, 2008.
11. Ł. Sosnowski and D. Ślęzak, "Networks of compound object comparators," In *Proceedings of the FUZZ-IEEE, India, Hyderabad 2013*, pp. 1–8.
12. Ł. Sosnowski, *Framework of Compound Object Comparators, Intelligent Decision Technologies*, 2015.
13. Ł. Sosnowski "Applications of comparators in data processing systems," *Technical Transactions, Automatic Control*, pp. 81–98, 2013.
14. Ł. Sosnowski, *Inteligentne dopasowanie danych przy użyciu teorii zbiorów rozmytych w systemach przetwarzania danych, Analiza systemowa w finansach i zarządzaniu*, pp. 214–218, 2009.
15. Ł. Sosnowski and D. Ślęzak, "Comparators for Compound Object Identification." in *Proc. of RSFDGrC 2011, ser. Lecture Notes in Computer Science*, vol. 6743, 2011, pp. 342–349.
16. Ł. Sosnowski and D. Ślęzak, "Fuzzy set interpretation of comparator networks," in *Pattern Recognition and Machine Intelligence - 6th International Conference, PREM I 2015, Warsaw, Poland, June 30 - July 3, 2015, Proceedings*, 2015, pp. 345–353. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-19941-2_33
17. Ł. Sosnowski, A. Pietruszka, and S. Łazowy, "Election algorithms applied to the global aggregation in networks of comparators," in *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems*, ser. *Annals of Computer Science and Information Systems*, M. P. M. Ganzha, L. Maciaszek, Ed., vol. 2. IEEE, 2014, pp. pages 135–144. [Online]. Available: <http://dx.doi.org/10.15439/2014F494>
18. Ł. Sosnowski, "Identification with compound object comparators - technical aspects," in *Techniki informacyjne teoria i zastosowania*, J. Hołubiec, Ed. IBS PAN, 2011, vol. 1, pp. 168–179.
19. Ł. Sosnowski, D. Ślęzak, *How to Design a Network of Comparators, Brain and Health Informatics*, pp. 389–398, 2013.
20. M. Szczuka, D. Ślęzak, *Feedforward Neural Networks for Compound Signals, Theoretical Computer Science*, vol. 412, no. 42, pp. 5960–5973, 2011.
21. D. Ślęzak, A. Janusz, *Ensembles of Bireducts: Towards Robust Classification and Simple Representation, FGIT 2011 in Conjunction with GDC 2011, Jeju Island, Korea*, pp. 64–77, December 2011
22. K. Trojanowski, *Metaheurystyki praktycznie*, wyd.2, Podręczniki WSISiZ, Wydawnictwo Wyższej Szkoły Informatyki Stosowanej i Zarządzania (WIT), 2008.
23. L. A. Zadeh, *Fuzzy Sets, Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.

LEARNING OF COMPOUND OBJECTS COMPARATORS

Abstract. The paper presents the algorithm of learning the reference set of objects and thresholding parameter responsible for quality of results in the compound objects comparators. Article contains a compendium of knowledge about the theory of compound objects comparators.

Keywords: compound objects comparators, similarity, recognition and identification of objects, machine learning, self organizing set of reference objects



