INSTYTUT BADAŃ SYSTEMOWYCH
POLSKIEJ AKADEMII NAUK

TECHNIKI INFORMACYJNE TEORIA I ZASTOSOWANIA

Wybrane problemy
Tom 2 (14)

poprzednio

ANALIZA SYSTEMOWA W FINANSACH I ZARZĄDZANIU

Pod redakcją
Andrzeja MYŚLIŃSKIEGO

Warszawa 2012

INSTYTUT BADAŃ SYSTEMOWYCH
POLSKIEJ AKADEMII NAUK

TECHNIKI INFORMACYJNE TEORIA I ZASTOSOWANIA

Wybrane problemy
Tom 2 (14)

poprzednio

ANALIZA SYSTEMOWA W FINANSACH I ZARZĄDZANIU

Pod redakcją
Andrzeja Myślińskiego

Warszawa 2012

Wykaz opiniodawców artykułów zamieszczonych w niniejszym tomie:

Dr hab. inż. Andrzej MYŚLIŃSKI, prof. PAN

Dr hab. inż. Ryszard SMARZEWSKI, prof. KUL

Dr hab. Dominik ŚLĘZAK

Prof. dr hab. inż. Andrzej STRASZAK

Prof. dr hab. inż. Stanisław WALUKIEWICZ

Dr hab. Adam WIERZBICKI

ISBN 9788389475442

Identyfikacja obiektów złożonych przy użyciu komparatorów

Łukasz Sosnowski

Studia Doktoranckie IBS PAN
e-mail:*l.sosnowski@dituel.pl*

Abstract.. Artykuł przedstawia podstawowe informacje o komparatorach obiektów złożonych w zastosowaniach do identyfikacji. Praca omawia zarówno aspekty teoretyczne jak i zastosowania praktyczne. Użyte przykłady pochodzą z praktycznych zastosowań komercyjnych.

Keywords: komparatory, zbiory rozmyte, analiza konturów, granulacja.

1 Wprowadzenie

Funkcjonowanie otaczającego nas świata oparte jest na umiejętności przetwarzania informacji oraz podejmowania decyzji na ich podstawie. Informacja obecna jest w każdym zachodzącym procesie. Może być różnie zapisywana (kodowana) oraz możemy wyróżnić różne jej nośniki. Wymiana informacji wiąże się z umiejętnością jej kodowania, dekodowania oraz transmisji [21]. Natomiast decyzje podejmowane są na podstawie pewnych informacji wejściowych (które mogą pochodzić z wielu źródeł) oraz ich synergii, agregacji lub dekompozycji.

Można powiedzieć, że świat bazujący na informacji składa się z obiektów, które mogą stanowić zarówno źródło jak i odbiorcę informacji. Obiekty, które występują w otaczającej nas przestrzeni można podzielić na obiekty złożone oraz obiekty proste. Obiektami prostymi nazywać będziemy atomowe, niepodzielne obiekty stanowiące "zwykłe" byty, które posiadają pewne cechy, lecz nie składają się z innych obiektów. Obiektami złożonymi będą natomiast obiekty posiadające pewną strukturę, mogące składać się z innych obiektów typu prostego lub złożonego.

Obiekty złożone jak i proste nie są bezpośrednio uczestnikami przetwarzania w systemach podejmowania decyzji. Przetwarzana jest ich pewna reprezentacja, stanowiąca najczęściej opis reprezentatywnej części obiektu. Reprezentacja ta może być różna dla tego samego obiektu w zależności od rodzaju procesu przetwarzania, celu i kontekstu. Na potrzeby wnioskowania o obiektach, istotne są relacje pomiędzy obiektami, w szczególności relacje podobieństwa czy porządku.

W wielu dziedzinach życia istnieje potrzeba łatwego porównywania oraz klasyfikacji obiektów, określania przynależności do grup lub zbiorów charakteryzujących się pewnymi z góry określonymi cechami. Potrzeba ta podyktowana jest koniecznością wykrywania cech obiektów, wykluczania pewnych zdarzeń, zapobiegania sytuacjom krytycznym, czy też identyfikacji na podstawie podobieństwa. Takie zapotrzebowanie generują

przeważnie systemy wyszukiwania informacji, systemy monitoringu, systemy rekomendacji, identyfikacji, moderacji i wiele innych, gdzie głównym punktem zainteresowania jest właśnie obiekt złożony i jego relacje z pewnymi wzorcami. W każdym przypadku mamy styczność z obiektami złożonymi, które są przetwarzane w celu osiągnięcia pewnego postawionego celu (różnego w zależności od danego systemu). Z pojęciem podobieństwa związany jest komparator, służący jako narzędzie do porównywania. Za jego pomocą możemy określić stopień podobieństwa pomiędzy porównywanymi obiektami.

Analizując występowanie, charakterystykę oraz specyfikę obiektów złożonych można zauważyć, iż ich przetwarzanie nierozłącznie wiąże się z podobieństwem, metodami porównywania cech i łączenia w zbiory. Wszytko to wskazuje na systemy wspomagania decyzji (Decision Support System) jako zastosowanie bezpośrednie dzielące się jednak na węższe dziedziny takie jak:

- **CBR (Case-based reasoning)** - systemy bazujące na gromadzonej wiedzy historycznej [15] (przypadkach rozwiązanych w przeszłości), w których nowe rozwiązania są konstruowane poprzez wyszukiwanie przypadków wcześniej pozytywnie zakończonych. Podstawowe etapy w procesie podejmowania decyzji w CBR to: a) znalezienie dobrego dopasowania zadanego (nowego) problemu z ewidencjonowanymi przypadkami historycznymi, b) dostosowanie poprzednich rozwiązań do aktualnego problemu, c) rozwiązanie zadanego problemu oraz zapis wyniku. W tym przypadku obiektem złożonym jest zadany problem, który na podstawie analizy porównawczej ze wspomnianymi elementami referencyjnymi jest do nich dopasowywany.
- **Multimedialne bazy danych** - systemy bazodanowe zdolne do przechowywania danych multimedialnych [2] takich jak zdjęcia, filmy, muzyka. Bazy takie poza ewidencjonowaniem mają za zadanie przetwarzać, wyszukiwać, agregować informacje o wspomnianych obiektach złożonych w sposób efektywny i akceptowalny dla użytkownika. Systemy te mają dostarczyć narzędzi do komunikacji z użytkownikiem, tak aby zadawane zapytania jak najlepiej wyrażały oczekiwania użytkownika. Dziedzina ta jest bezpośrednim rozszerzeniem klasycznych relacyjnych baz danych (choć nie tylko), gdzie pojęcie obiektu złożonego utożsamiane było bardziej z jego opisem znajdującym się w postaci krotek relacji opisanych pewnymi atrybutami. W przypadku baz multimedialnych metadane o obiektach wzbogacone zostały o same te obiekty (struktury zamknięte), które muszą być przez silniki baz danych porównywane, szeregowane na podobnych zasadach jak dane w standardowych bazach relacyjnych.
- **CEP (Complex event processing)** - na przykład systemy monitoringu zdarzeń w czasie rzeczywistym stosowane w przemyśle. Systemy te analizują dane z różnych źródeł, porównują je i wykrywają wzorce, trendy i wyjątki związane z analizowanymi zdarzeniami [11]. Dzięki temu niepożądane sytuacje mogą zostać wykryte zanim nastąpią. Umożliwia to podjęcie odpowiednich decyzji w celu ich uniknięcia. W tej dziedzinie obiektami złożonymi są konfiguracje obserwacji na zbiorach sensorów jak również wzorce i trendy, z którymi porównywane są informacje pozyskane.
- **Silniki wyszukiwania** - silniki wyszukiwania [3] informacji różnego typu, które w większości przypadków pracują na danych zaszumionych. Zadawane zapytania są niejednokrotnie nieprecyzyjnie oraz niejednoznacznie zakodowane. Systemy oparte o te silniki optymalizują trafność zwróconych wyników do zadanego zapytania, w celu dostarczenia najbardziej pożądanej informacji dla użytkownika. Obiekty zło-

żone mogą przyjmować tutaj postać różnego rodzaju dokumentów tekstowych indeksowanych przez systemy wyszukiwania (np. strony html), ale również mogą to być obrazy, dźwięki, filmy etc. Obiekty te w zależności od typu są poddawane analizie w celu określenia dodatkowego opisu wynikającego z ich struktury lub zawartości. Przykładem może być analiza histogramowa tekstu, gdzie podstawowym zadaniem jest policzenie rozkładu występowania słów w tekście (z pominięciem słów nieistotnych) i na tej podstawie wnioskowanie przynależności dokumentu do ustalonych grup (kategorii).

We wszystkich dziedzinach wymienionych powyżej występuje wspólna potrzeba badania podobieństwa. W przykładach tych nie zostało sprecyzowane, w jaki sposób podobieństwo jest badane. W różnych dziedzinach, systemach i metodach podobieństwo może być definiowane na różne sposoby. Jednakże w każdym z tych przypadków można myśleć o uniwersalnej metodyce opartej na zastosowaniu komparatorów [13] do porównywania obiektu złożonego ze zbiorem obiektów referencyjnych. Z przytoczonych wcześniej przykładów wynika, iż obiekty złożone są zróżnicowane pod względem typu, zastosowań oraz istotności. Niezależnie jednak od tych różnic, proponowana metodyka zapewnia wspólne podejście do każdego z tych przypadków. Dzięki temu rozwój technik porównawczych udoskonala badanie obiektów na wielu polach eksploatacji jednocześnie.

Proponowana w niniejszej pracy metodyka zakłada istnienie pewnego zbioru referencyjnego (lub hierarchii zbiorów referencyjnych), w którym znajdują się dobrze już znane przeanalizowane bądź opisane obiekty. Innymi słowy, są to obiekty pogrupowane względem pewnych cech, lecz reprezentujące różne wartości danej cechy (zbiór reprezentatywny). Przykładem zbioru referencyjnego może być zbiór map administracyjnych danego terenu, np. Polski. Oczywiście zbiór ten może mieć również hierarchię rozpoczynającą się od zbioru ogólnego map, następnie podziału po kontynentach, krajach itd.

Podstawową ideą proponowanej metodyki jest znalezienie obiektu referencyjnego (lub wielu obiektów) najbardziej podobnego do zadanego obiektu na wejściu. Zakładając, iż podobieństwo spełnia pewne minimalne wymogi co do jego jakości, próbujemy wykorzystać posiadaną wiedzę o obiekcie referencyjnym do wnioskowania rozwiązania (może to być identyfikacja obiektu, przypisanie do grupy, etc.) dla zadanego obiektu. Następnie wykonujemy badanie podobieństwa obiektu wejściowego ze względu na różne cechy przy użyciu tego samego zbioru referencyjnego. Kolejnym krokiem jest łączenie wyników i podejmowanie decyzji na podstawie wielu kryteriów. Możliwe jest też badanie tego samego obiektu względem różnych zbiorów referencyjnych. Przy takim podejściu obiekt złożony jest dekomponowany na obiekty prostsze, a następnie poszczególne składowe poddawane są procesowi porównywania z różnymi (dobranymi na etapie wstępnej identyfikacji) zbiorami referencyjnymi. Przy doborze zbiorów referencyjnych korzystamy z hierarchii tych zbiorów (jeśli jest to możliwe w danym przypadku). Różnica pomiędzy podanymi podejściami polega na metodzie łączenia wyników. Schemat pierwszy wydaje się być łatwiejszy do konstrukcji ze względu na tożsamość zbioru referencyjnego, co umożliwia dokonywanie różnego rodzaju agregacji rankingów uzyskanych przy badaniu poszczególnych cech. Drugi schemat jest również możliwy do implementacji np. przy użyciu reguł (w tym również rozmytych), gdzie wyniki poszczególnych komparatorów działających na zdekomponowanych obiektach stanowią przesłanki zasilające reguły.

Ważnym założeniem proponowanej metodyki jest jej uniwersalność. Przede wszystkim skala oceny podobieństwa dla każdego rodzaju i typu obiektu powinna być taka

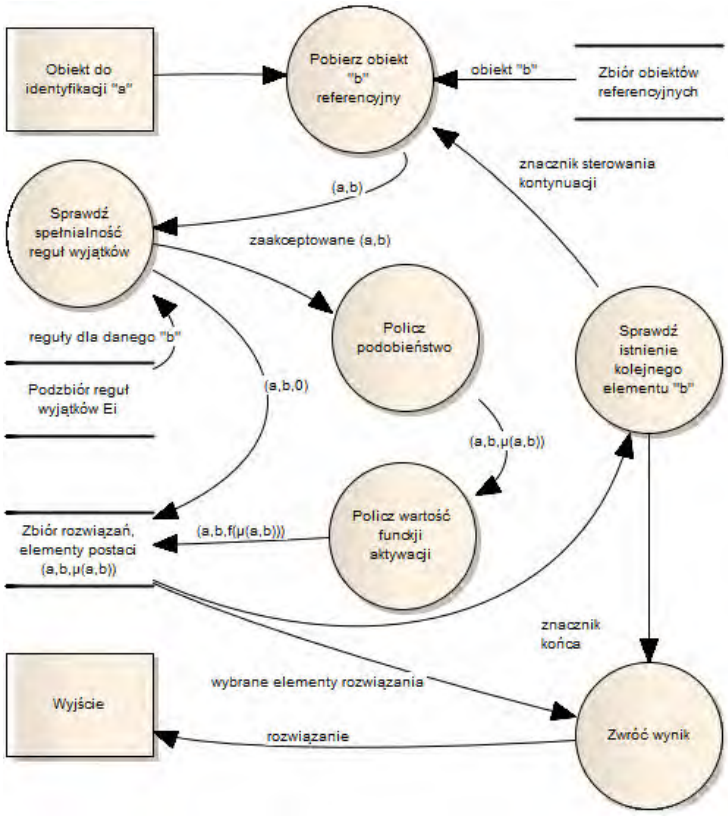
sama. Nie powinna być przy tym binarna, powinna raczej określać w dowolnie dokładny sposób stopień podobieństwa. Ustandaryzowana forma zwracania wyników porównania ułatwia łączenie wyników osiągniętych na różnych polach przetwarzania informacji. Ze względu na duży stopień złożoności obiektów, czasami trudno skonstruować formułę używaną do porównywania i w stu procentach spełniającą wymagania wszystkich przypadków. Dlatego też wprowadzony został mechanizm definiowania wyjątków, który umożliwia uproszczenie formuł i zapisanie pewnej wiedzy w postaci reguł, których spełnienie oznacza eliminację danego rozwiązania. Podobne mechanizmy możemy obserwować w innych dziedzinach (np. układy cyfrowe), gdzie występują tzw. dyskryminatory, które pozwalają wykluczyć pewne rozwiązania na podstawie pewnych wartości wejściowych.

Analizując przedstawioną metodykę możemy zauważyć, że stawia ona pewne wymogi do spełnienia na etapie realizacji. Przede wszystkim wymaga usystematyzowanej wiedzy i umiejętności jej zapisu w postaci obiektów referencyjnych pogrupowanych w zbiory w pewnej hierarchii (choć nie zawsze). Obiekty referencyjne wymagają dobrej reprezentacji, tworzonej efektywnie i dającej szerokie możliwości przetwarzania. Jako że podstawowym elementem schematu przetwarzania jest komparator, dlatego operacje wykonywania szybkich porównań będą kluczowe dla efektywnej realizacji. Te trzy podstawowe wymogi muszą być spełnione, aby metodyka była przydatna w klasie systemów DSS (Decision Support System). Dlatego też dobór odpowiednich narzędzi implementacyjnych ma kluczowe znaczenie dla powodzenia przedsięwzięcia. Uniwersalny schemat budowy komparatora obiektów złożonych oparty jest o teorię zbiorów i relacji rozmytych [38,9]. Wyposażony w mechanizmy kontroli jakości rozwiązania oraz wykluczania z góry pewnych rozwiązań. Z punktu widzenia ogólnie pojętych systemów przetwarzania danych możliwe jest zastosowanie różnych rozwiązań technicznych i implementacyjnych. W przedstawione rozwiązanie, opiera się na systemach baz danych [7] i hurtowni danych [36], jako najpowszechniejszych narzędziach spotykanych przy systemach przetwarzania informacji. Warto jednak zaznaczyć, iż nie jest to jedyne rozwiązanie i są możliwe inne implementacje równie efektywne w pewnych określonych warunkach.

W niniejszym rozwiązaniu hurtownie danych stanowią część odpowiedzialną za narzędzia, z których będę korzystał do budowy zbiorów referencyjnych, jak również całego systemu przetwarzania łączącego ewidencję obiektów z agregacją osiągniętych wyników. Dzięki temu w pracy czytelnik znajdzie teoretyczny zarys metodyki oraz aspekty techniczne praktycznej realizacji przy użyciu pewnego zestawu narzędzi. W pracy znajdziemy również przykład obrazujący realne zastosowanie metodyki do identyfikacji map konturowych.

2 Komparatory obiektów złożonych

Zdefiniujmy jednostkę składającą się z kilku podstawowych elementów: rodziny zbiorów reguł wyjątków, komparatora rozmytego zdolnego porównać dwa obiekty, zbioru obiektów referencyjnych, funkcji aktywacji rozwiązania oraz metody zwracania wyników. Elementy te są ze sobą powiązane w ściśle określony sposób przedstawiony na diagramie przepływu danych na rysunku nr 2. Jednostka ta jest bytem złożonym, umożliwiającym określenie podobieństwa jednego obiektu do poszczególnych obiektów zbioru referencyjnego i w zależności od metody zwracania wyników wyboru najlepszego rozwiązania względem przyjętego kryterium (badanej cechy) [24].



Rysunek 1. Diagram przepływu danych komparatora - schemat ogólny

Taka budowa zapewnia kilka innowacyjnych funkcjonalności względem dotychczas stosowanych komparatorów. Tak jak zostało wspomniane w rozdziale 1, komparatory te pracują z obiektami (pewnymi ich reprezentacjami) a nie jedynie z liczbami reprezentującymi pewne wartości badanej cechy. Inaczej mówiąc, omawiane komparatory nie pracują z gotowym wektorem cech, lecz same taki wektor współtworzą. Takie podejście daje możliwość zastosowania technik aproksymacji w stosunku do reprezentacji obiektu (np. przy użyciu technik granulacji [16] i wyliczania pewnych wartości charakterystycznych). Praca z obiektami daje możliwość większego uogólnienia, gdyż nie wymaga bardzo drobiazgowego przetwarzania wstępnego celem uzyskania wartości wejściowej do komparatorów klasycznych. Tutaj komparator jest odpowiedzialny za podejmowanie decyzji w części której dotyczy, czyli np. w obrębie przetwarzania danej cechy. Uogólnienie danych na wejściu (obiekt złożony) przenosi się na uogólnienie procesu decyzyjnego, tzn. im obiekt jest bardziej złożony tym podejmowana decyzja może być bardziej znacząca, np. komparator pracujący na tekście podejmuje decyzje klasyfikujące całe teksty do grup. Aby uzyskać ten sam efekt przy użyciu komparatorów klasycznych, musielibyśmy dokonać wielu operacji na zewnątrz komparatora i prawdopodobnie użyć bardzo wielu kompa-

ratorów cyfrowych. Różnica więc jest taka, iż komparator obiektów złożonych scala w większe części przetwarzanie i podejmuje decyzje wyższego stopnia w hierarchii, natomiast komparatory klasyczne ograniczają się do pewnych operacji atomowych wchodzących w skład innych układów cyfrowych. Kolejną bardzo istotną cechą jest wykorzystanie komparatora rozmytego, zapewniającego konkretny wynik podobieństwa między obiektami wyrażony liczbowo a nie jedynie określenie znaku tej relacji. To jest w pewnym sensie powrót do pierwotnych idei przytoczonych w [13]. Nieodłącznym elementem tego komparatora jest zbiór referencyjny obiektów, jak również zbiorów reguł wyjątków. Te dwa elementy pozwalają na dołączenie mechanizmów uczących [4], które poprzez modyfikacje zawartości tych zbiorów mogą wpływać na przyszłe wyniki komparatorów.

2.1 Podstawowe własności komparatora obiektów złożonych

Komparatory obiektów złożonych są wielowarstwową strukturą składającą się z kilku komponentów ściśle ze sobą powiązanych. Każdy z tych komponentów indywidualnie spełnia pewne własności, lecz w połączeniu ze sobą redukują wspólne własności do dwóch podstawowych lecz bardzo istotnych.

Komparator obiektów złożonych oznaczamy poprzez K, gdzie $K(a, B)$, gdzie oznacza wartość działania komparatora, a jest obiektem badanym należącym do zbioru obiektów badanych A, natomiast B jest zbiorem obiektów referencyjnych. Będziemy również wyróżniali funkcję charakterystyczną dla danego komparatora, związaną z metodą selekcji danych wynikowych, tzn. w jaki sposób wybierane są rezultaty do zwrócenia poprzez komparator. W praktyce często będzie to funkcja maksimum, lecz również funkcja sortująca elementy zbioru np. sort. Przy tak zdefiniowanych oznaczeniach komparatory obiektów złożonych powinny spełniać następujące własności:

1.
$$K(a, \varnothing) = \varnothing, \forall a \in A \tag{1}$$

2.
$$K(a, a) = a \vee \varnothing, \forall a \in A \tag{2}$$

3.
$$K(a, B) = f_{ch}(K(a, B_1) \cup K(a, B_2)), \forall B_1, B_2 : B_1 \cap B_2 = \varnothing \wedge B_1 \cup B_2 = B \tag{3}$$

gdzie f_{ch} jest funkcją charakterystyczną komparatora definiującą metodę zwracania wyników (np. MAX, SORT, etc).

Pierwsza własność reguluje szczególny przypadek, w którym zbiór referencyjny jest pusty. Druga definiuje możliwe wyniki działania komparatora przy zbiorze referencyjnym jednoelementowym którego jedynym elementem jest obiekt badany. Własność ta zachodzi niezależnie od zawartości reguł wyjątków. Trzecia własność umożliwia podział zbioru referencyjnego na rozłączne podzbiory i wykonanie na nich przetwarzania. Dzięki tej własności możemy zrównoleglić przetwarzanie (wiele podzbiorów przetwarzać w jednym czasie) w celu optymalizacji czasu przetwarzania, jednocześnie gwarantując uzyskanie tego samego wyniku co dla przetwarzania całego zbioru. Uzyskane wyniki należy poddać działaniu pewnej funkcji charakterystycznej dla danego komparatora, która wyznacza porządek w zbiorze wyników. Oznacza to, iż wyniki uzyskane z przetwarzania

poszczególnych podzbiorów są policzone i wyznaczone tak samo jak dla jednego zbioru przetwarzania, należy jedynie je uporządkować.

Dla pewnych szczególnych konfiguracji zbioru wyjątków możemy rozważać także inne własności komparatorów, np. K(a,b) = K(b,a), które mogą być przydatne w pewnych szczególnych przypadkach.

2.2 Szczegóły techniczne

Do określenia podobieństwa wybranych obiektów obliczamy funkcję przynależności do relacji (komparator rozmyty). W zależności od klasy obiektów lub badanej cechy, ta funkcja będzie dobierana tak, aby najlepiej mierzyła podobieństwa obiektów dla zadanych parametrów. Zgodnie z definicją relacji rozmytych [9], jest to funkcja dla której wartości skrajne przeciwdziedziny wskazują odpowiednio na całkowity brak podobieństwa („0") lub całkowite podobieństwo („1"). Cechy zakazane zdefiniowane są w zbiorze R takim, że dla każdego elementu zbioru referencyjnego istnieje podzbiór reguł R. Podzbiór ten może być zbiorem pustym lub posiadać zdefiniowane elementy (reguły) stanowiące o wykluczeniu podobieństwa pomiędzy obiektami w przypadku spełnienia jednej z reguł. Jeśli reguła jest spełniona to badanie podobieństwa traci sens i przechodzimy do wyboru kolejnego obiektu referencyjnego. Następnie będziemy rozważać funkcję aktywacji, która będzie określała minimalną jakość naszego rozwiązania. Podobnie jak w przypadku funkcji przynależności, funkcja aktywacji - a dokładnie jej parametr "p" - będzie indywidualnie dopasowywany do konkretnej klasy obiektów lub nawet do konkretnych obiektów referencyjnych. Do jego wyboru może być użyta wiedza ekspercka. Można również użyć algorytmów ewolucyjnych [20,1] w celu próby optymalizacji doboru parametru (tak aby "p" był jak najmniejszy, a jednocześnie gwarantował dobrą jakość rozwiązania). Realizacja klasyfikacji cech zakazanych została zaplanowana dla systemu regułowego. Nie wyklucza się tutaj również reguł rozmytych, dzięki czemu będzie można rozpatrywać nieostre występowanie cech określanych przez reguły. W końcowej fazie przetwarzania następuje realizacja procesu zwrócenia wyników. Proces ten jest realizowany w zależności od typu komparatora. Stosowne różnice zostaną pokazane w dalszej części artukułu. Zanim omówię różnice, opiszę poszczególne części składowe komparatora obiektów złożonych.

Zbiór obiektów referencyjnych Zbiór obiektów względem których dokonujemy porównania. Są to obiekty o których posiadamy pewną wiedzę pozwalająca na zaklasyfikowanie do jednego zbioru. Elementy te są dobrane względem pewnej cechy lub wielu cech wspólnych, choć niekoniecznie podobnych do siebie (przykład cechy kolor: obiekty posiadają kolor lecz wartości tych kolorów mogą być różne). Każdy element referencyjny posiada odpowiedni podzbiór reguł wyjątków zawarty w zbiorze reguł wyjątków R.

Zbiór reguł wyjątków Zdefiniujmy zbiór wszystkich możliwych reguł R oraz podzbiory R_j indeksowane poprzez "j" takie że dla każdego elementu b_j zbioru referencyjnego istnieje podzbiór reguł R_j, który jest zbiorem pustym lub zawiera elementy stanowiące reguły mające zastosowanie do badanego elementu "a" w przypadku porównywania z "b".

Reguła może być opisana poprzez zespół predykatów logicznych połączonych operatorami koniunkcji, negacji lub alternatywy. Jeśli dane zdanie logiczne jest prawdziwe wtedy dana reguła zostaje uznana za spełnioną.

Komparator rozmyty Komparator w rozumieniu definicji z punktu 2. Komparator określa cechę, którą bada cały komparator złożony. Dobór komparatora rozmytego to jeden z najważniejszych czynników stanowiący o skuteczności ogólnej jednostki. To element odpowiedzialny za definiowanie podobieństwa w ogólnej postaci.

Funkcja aktywacji rozwiązania Funkcja zdefiniowana jako

$$f(\mu(a,b)) = \begin{cases} 0 & \mu(a,b)<p \\ \mu(a,b) dla & \mu(a,b)\geq p \end{cases} \tag{4}$$

Funkcja ma za zadanie ograniczenie zbyt słabych rozwiązań, tzn. takich, dla których podobieństwo jest zbyt małe. Jeśli wartość parametru p nie zostanie osiągnięta, wtedy wartość podobieństwa wynosi "0".

Metoda zwracania wyników Metoda określająca w jakiej postaci otrzymamy wyniki końcowe. Od doboru tej metody zależy rodzaj użytego komparatora obiektów złożonych.

Wyróżniamy kilka rodzajów komparatorów obiektów złożonych. Każdy z nich ma bardzo podobną budowę. Różnice polegają na postaci, w jakiej zwracane są wyniki oraz jak wyniki te zostały uzyskane. Różnice pomiędzy komparatorami nie są znaczące w schemacie i implementacji, jednakże ich złożoności obliczeniowe mogą się znacznie różnić. W dalszej części artykułu został zdefiniowany podział wg. typów komparatorów obiektów złożonych.

2.3 Model identyfikacji obiektów

Przykładem zastosowania proponowanej metodyki jest budowa modelu do identyfikacji obiektów złożonych. Model posiada trzy podstawowe fazy działania:

- **Akwizycja danych** - etap, w którym pozyskujemy obiekty do badania z innych obiektów złożonych. W wyniku wykonania tego etapu powstaje zbiór elementów A gotowy do dalszej obróbki przybliżającej do osiągnięcia głównego celu. Przykładem akwizycji, może być segmentacja tekstu oraz selekcja słów istotnych z punktu widzenia próby klasyfikacji do kategorii. Innym przykładem jest segmentacja obrazu celem uzyskania informacji o obiektach znajdujących się na tym obrazie.
- **Przetwarzanie wstępne** - służy do pozyskiwania informacji o obiekcie, budowy reprezentacji obiektu, w niektórych przypadkach zmniejszenia jego złożoności poprzez techniki granulacji informacji [16]. Faza ta jest ważnym elementem pozwalającym na przystosowanie obiektów do znanych technik przetwarzania oraz uzyskania efektywnego sposobu obliczeń. Etap ten przygotowuje dane o obiektach, które stanowią bezpośrednio wejście dla komparatorów. W większości przypadków faza ta jest wykonywana dla każdego komparatora niezależnie, gdyż przeważnie poszczególne komparatory wymagają zupełnie różnych reprezentacji obiektów. Z drugiej strony

doświadczenia opisane w moich wcześniejszych publikacjach ukazują, że dla różnych rodzajów obiektów można stosować analogiczne, choć różnie skonfigurowane narzędzia przetwarzania.

– **Porównywanie** - faza w której komparatory działają bezpośrednio na reprezentacjach obiektów zarówno referencyjnych jak i badanych. W fazie tej może działać wiele komparatorów obiektów złożonych najczęściej w sposób współbieżny. Oczywiście można również wyobrazić sobie analogię do wielowarstwowych sieci neuronowych jednokierunkowych [14], gdzie istnieją tzw. warstwy neuronów. Tutaj również można by rozpatrywać warstwy komparatorów, które uzyskiwałyby dane na wejście jako wynik komparatora z warstwy poprzedniej. Dotychczas jednak stworzone i stosowane modele zakładają architekturę jednowarstwową, aczkolwiek wielo-komparatorową.
– **Łączenie wyników** - proces agregacji poszczególnych wyników jednostkowych pochodzących z komparatorów w jedną finalną decyzję [26]. W zależności od rodzaju użytych komparatorów oraz zbiorów referencyjnych które zostały użyte dla poszczególnych jednostek porównawczych, metoda łączenia wyników może ulegać zmianie. Działając w oparciu o ten sam zbiór obiektów referencyjnych (choć niekoniecznie te same reprezentacje) najstosowniejszym wydaje się rozwiązanie oparte na rankingach. Każdy komparator tworzy swoisty ranking poprzez zwrócenie zbioru z porządkiem liniowym, wyznaczanym przez funkcję przynależności do relacji. Najprostszym rozwiązaniem wydaje się połączenie rankingów metodami uśredniania lub ważonymi w zależności od istotności cech dla danego modelu.

W przypadku przetwarzania wielu obiektów wejściowych i przy pewnych dodatkowych założeniach, model może samodzielnie poprawiać uzyskane rozwiązania poprzez wprowadzenie konkurencyjności między obiektami [33]. Poprzez to uzyskujemy zależność wyniku jednego obiektu od wyników pozostałych obiektów. Procedura ulepszania rozwiązania zakłada, iż na wejściu mamy zbiór obiektów i dokonujemy identyfikacji kolejnych dowolnie wybranych elementów tego zbioru bez powtórzeń. Dodatkowo zakłada się, iż funkcja podobieństwa jest różnowartościowa. Założenie to jest istotą tej dodatkowej procedury. Dzięki tej własności przekształcenia, jesteśmy w stanie zauważyć przypadki, które muszą być poddane dalszemu badaniu, właśnie poprzez naruszenie postawionego wyżej założenia.

	ref1	ref2	ref3	ref4	ref5
obj1	**0,91**	0,65	0,23	0,45	0,49
obj2	**0,82**	0,81	0,34	0,42	0,12
obj3	0,23	0,12	**0,88**	0,23	0,31
obj4	0,51	0,43	0,32	**0,94**	0,12
obj5	0,49	0,21	0,14	0,39	**0,89**

⇒

	ref1	ref2	ref3	ref4	ref5
obj1	**0,91**	0,65	0,23	0,45	0,49
obj2		**0,81**	0,34	0,42	0,12
obj3	0,23	0,12	**0,88**	0,23	0,31
obj4	0,51	0,43	0,32	**0,94**	0,12
obj5	0,49	0,21	0,14	0,39	**0,89**

Rysunek 2. Metoda współzawodnictwa wyników ulepszająca efektywność modelu

Po przeprowadzeniu obliczeń we wszystkich fazach modelu dla każdego obiektu wejściowego otrzymujemy macierze rozwiązań dla poszczególnych komparatorów. Dla ustalenia uwagi przyjmijmy, iż indeks i w tej macierzy $[x_{ij}]$, dotyczy obiektów wejściowych zaś j obiektów referencyjnych. Wartością x_{ij} jest podobieństwo pomiędzy danym obiektem wejściowym a obiektem referencyjnym. Jeśli kilka obiektów wejściowych wskazuje

najwyższe, niezerowe podobieństwo do tego samego obiektu referencyjnego, to wiersze te zostają zakwalifikowane do dodatkowego przetwarzania. Spośród tych zakwalifikowanych wierszy pozostawiamy bez zmian jedynie ten, który ma najwyższą wartość podobieństwa dla "wspólnego" obiektu referencyjnego (jeśli jest więcej niż jeden wiersz o tej samej wartości maksymalnej podobieństwa to wtedy dokonujemy wyboru w inny wybrany sposób: losowo, poprzez badanie odległości danego rozwiązania od następnego, etc). Dla pozostałych wierszy podobieństwo do tego konkretnego obiektu referencyjnego jest zerowane. Powoduje to, iż dla tego konkretnego obiektu wejściowego inny obiekt referencyjny staje się rozwiązaniem o maksymalnej wartości podobieństwa. Wtedy ponownie należy sprawdzić spełnialność założenia różnowartościowości. Procedurę tą powtarzamy aż do uzyskania rozłącznych rozwiązań (dany obiekt referencyjny dopasowany jest do dokładnie jednego obiektu wejściowego).

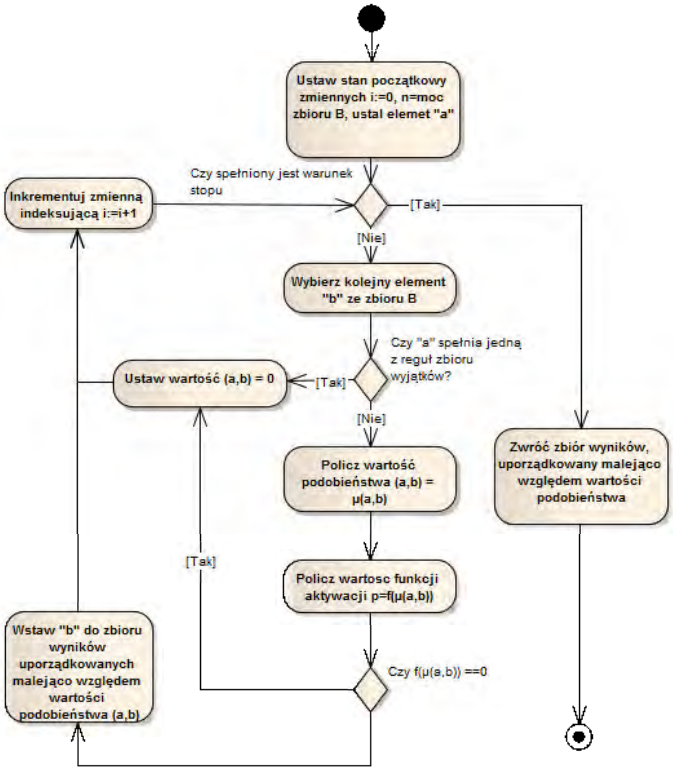
2.4 Klasyfikacja komparatorów obiektów złożonych

W celu łatwiejszego użycia komparatorów obiektów złożonych wprowadziłem ich podział ze względu na pewne charakterystyczne cechy. Poszczególne opisywane komparatory działają w oparciu o wspólny schemat budowy. Różnią się w zakresie momentu przerwania przetwarzania, jak i jakościowych parametrów mających wpływ na rozwiązanie.

Typ standardowy [K_{st}] Podstawowym typem komparatora jest taki, który w wyniku przetwarzania zwróci uporządkowany zbiór obiektów referencyjnych z przypisaną wartością podobieństwa dla obiektu badanego. W tym przypadku zakłada się sprawdzenie podobieństwa obiektu wejściowego z każdym elementem zbioru referencyjnego.

Komparator standardowy zwraca zbiór o elementach postaci $(a, b, \mu(a, b))$, dla $a \in A$, $b \in B$, gdzie A jest zbiorem obiektów do przetworzenia, a B jest zbiorem referencyjnym. Jego funkcją charakterystyczną jest funkcja porządku, która dokonuje uporządkowania elementów malejąco względem wartości podobieństwa. Jednocześnie jest to komparator, którego złożoność oczekiwana jest bliska złożoności pesymistycznej gdyż zawsze dokonywane jest przetwarzanie dla wszystkich elementów zbioru. Jedyne różnice mogą powstać przy różnych konfiguracjach zbiorów reguł wyjątków. Charakterystyczne jest to, że zwracany ciąg ma dokładnie tyle elementów ile elementów posiada zbiór B.

Typ MAX [K_{max}] Inny wyróżniony typ komparatorów obiektów złożonych opiera się na wybraniu tylko tych par, których wartość zwracana przez komparator rozmyty jest największa. Oczywiście wartości największych może być w szczególności tyle samo ile elementów zbioru referencyjnego. Wtedy wyniki będą takie same jak dla komparatora typu standardowego. Jednakże w innym przypadku liczba elementów będzie mniejsza. Komparator może mieć zastosowanie w przypadku, gdy interesują nas jedynie najbardziej podobne pary obiektów. Budowa wewnętrzna komparatora w większości jest identyczna z budową komparatora standardowego. Różnice powstają jedynie w końcowej fazie, w której rozróżniamy przypadek równości wartości zwracanej przez komparator rozmyty i dodatkowo jest on obsługiwany oraz może występować inny warunek stopu, który może zapewnić lepszą efektywność.



Rysunek 3. Diagram aktywności komparatora obiektów złożonych - typ standardowy

Typ singleton [K_{sgn}] Singleton jest specyficzną odmianą komparatora typu MAX. W tym jednak przypadku będziemy rozpatrywali zbiór wyników jednoelementowy. Możliwymi wynikami są albo zbiór pusty, albo zbiór zawierający element postaci $(a, b, max(\mu(a, b)))$. Różnica jednak jest taka, że w tym przypadku wybieramy tylko jeden taki element. W zależności od tego który z elementów wybierzemy, będziemy rozpatrywali jeszcze dodatkowe podtypy komparatora obiektów złożonych tego typu.

Max First *[K_{smf}]* Komparator typu singleton, w którym bierzemy pod uwagę pierwszy element maksymalny. W przypadku wystąpienia identyczności ten komparator optymalizuje przetwarzanie, gdyż kolejne porównania nie są już wykonywane. Wynika to z faktu, iż większej wartości od 1 nie możemy uzyskać przy badaniu podobieństwa tą metodą. A zatem jest to pierwszy element maksymalny, czyli wynikowy. Ten typ komparatora warto stosować w obszarach, w których będziemy badać podobieństwo pojedynczej cechy. W przypadku bardziej złożonych problemów i większej liczby badanych cech często potrzebujemy odwołać się do pewnego podzbioru uzyskanych wyników a nie jedynie wyniku z maksymalną wartością podobieństwa. Wynika to między innymi z różnej wagi cech oraz różnej skuteczności komparatorów działających równolegle.

Max Last *[K_{sml}]* Analogiczny typ komparatora różniący się warunkiem stopu przetwarzania oraz zbiorem zwracanych wyników. Komparator zwraca zbiór zawierający obiekt referencyjny dla którego wartość podobieństwa z danym obiektem "a" była maksymalna i który został pobrany ze zbioru referenyjnego do badania jako ostatni (wsród obiektów dla których występuję najwyższa wartość podobieństwa). Komparator może mieć zastosowanie w przypadkach, gdzie stosujemy uporządkowane zbiory referencyjne względem pewnej preferencji wyboru obiektów (tzn. takie gdzie kolejne obiekty referencyjne pobierane do przetwarzania są bardziej preferowane jako wynik niż poprzednie). Praktyczna realizacja może być wykonana w systemach moderacji tekstów, gdzie zbiorem referencyjnym mogą być słowa zakazane, lecz uporządkowane względem przynależności do pewnych klas, które mogą wyznaczać dalszy tok działania. Jeśli badany obiekt posiada elementy z klasy "całkowicie niedopuszczalne", wtedy cały tekst może zostać nieopublikowany. Jeśli natomiast ostatnim elementem referencyjnym byłby element należący do klasy "dopuszczalne z uwagami" wtedy tekst mógłby być opublikowany lecz z pewnym dopiskiem lub uwagą.

Typ "Q" [K_q] Ten typ komparatora jest podobny to typu standardowego, jednakże zwracany jest zbiór uporządkowany tylko tych elementów dla których spełniona jest dodatkowa zależność $\mu(a,b) \geq q$, gdzie $q \geq p$ oraz $q \leq 1$. Typ ten można również rozpatrywać w dwóch wariantach podtypów, kiedy interesuje nas zbiór wynikowy jednoelementowy lub zbiór pusty.

Q First *[K_{qf}]* W tym przypadku interesuje nas jedynie pierwszy wynik, który przekroczy wartość zadaną "q" (oczywiście przy zachowaniu wcześniej zdefiniowanych warunków). Typ ten jest efektywniejszy ze względu na lepszą oczekiwaną złożoność obliczeniową. Zakładamy tu pewien próg, którego przekroczenie spełnia nasze oczekiwania wobec wyniku (choć niekoniecznie jest to wartość maksymalna).

Q Last *[K_{ql}]* Typ analogiczny do poprzednio opisywanego, jednakże o zamienionej kolejności zwracanego elementu co może mieć wpływ na wydajność działania. Tutaj przetwarzanie w większości przypadków wykona się dla każdego elementu referencyjnego, gdyż dopiero ostatni element może stanowić wynik. Bywają jednak odstępstwa od tej reguły dla pewnych specyficznych obiektów oraz zbiorów referencyjnych.

Typ "co najwyżej n" [K_{topN}] Ten rodzaj komparatora zwraca jako wyniki zbiór uporządkowany elementów referencyjnych o liczności co najwyżej "n", których wartość podobieństwa do obiektu "a" jest najwyższa. Typ ten jest pochodną wyżej opisywanych typów lecz dzięki swej właściwości jest w pewnych przypadkach bardziej efektywny. W zależności od badanego problemu możemy sterować parametrem "n" skracając w ten sposób czas przetwarzania i ograniczając jednocześnie liczbę wyników, które będą miały wpływ na podejmowanie finalnej decyzji.

Jak widać poszczególne komparatory różnią się szczegółami implementacji, które z punktu widzenia algorytmu różnią się nieznacznie. W celu łatwiejszego porównania poszczególnych komparatorów została przygotowana tabela nr 1.

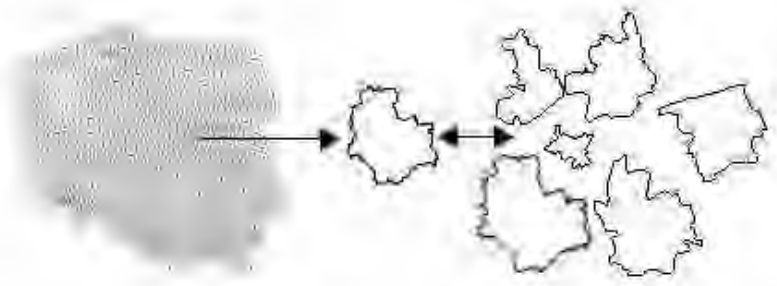
Typ komparatora	Zwracanie wyników	Warunek stopu
K_{st}	wszystkie obiekty referencyjne	przetworzenie wszystkich obiektow referencyjnych
K_{max}	tylko obiekty o najwyższej wartości podobieństwa	znalezienie podzioru obiektów o maksymalnym podobieństwie
K_{sgn}	jeden obiekt o maksymalnym podobieństwie	znalezienie założonego obiektu o maksymalnym podobieństwie
K_{smf}	pierwszy obiekt o maksymalnym podobieństwie	znalezienie pierwszego obiektu o maksymalnym podobieństwie
K_{sml}	ostatni obiekt o maksymalnym podobieństwie	znalezienie ostatniego obiektu o maksymalnym podobieństwie
K_q	podzbiór obiektów o podobieństwie nie mniejszym niż “q”	przetworzenie wszystkich obiektow referencyjnych
K_{qf}	pierwszy obiekt o podobieństwie nie mniejszym niż “q”	znalezienie pierwszego obiektu o podobieństwie nie mniejszym niż “q”
K_{ql}	ostatni obiekt o podobieństwie nie mniejszym niż “q”	znalezienie ostatniego obiektu o podobieństwie nie mniejszym niż “q”
K_{topN}	podzbiór “n” elementowy obiektów z najwyższymi wartościami podobieństw	wybranie “n” elementów o najwyższych wartościach podobieństwa

Tablica 1. Zestawienie typów komparatorów

3 Model identyfikacji map konturowych

Model identyfikacji map konturowych podziału administracyjnego Polski powstał na potrzeby komercyjnego projektu Wizualizacji Wyborów Samorządowych w Polsce w 2010 roku [27]. Jego celem jest umożliwienie identyfikacji obszarów znajdujących się na mapie konturowej Polski z podziałem terytorialnym na powiaty, miasta na prawach powiatu lub gminy. Przykład dotyczy jedynie podziału Polski, lecz może być użyty dla dowolnych obszarów czy innych obiektów reprezentowanych w sposób konturowy.

Identyfikacja w tym kontekście oznacza przypisanie kodu terytorialnego GUS, który umożliwi wizualizację danych dla tego obszaru (np. danych o frekwencji). Do dyspozycji mamy zbiory referencyjne, których elementami są mapki konturowe pojedynczych obszarów, o których mamy pełną potrzebną wiedzę. Ideę działania przedstawia rysunek nr 4.



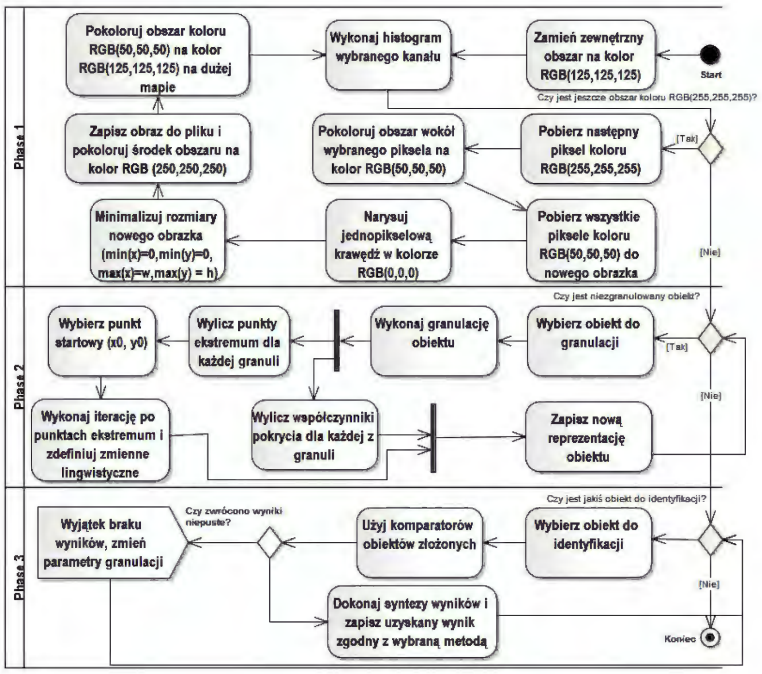
Rysunek 4. Idea działania systemu identyfikacji

Rysunek nr 5 przedstawia diagram aktywności dla przykładowego modelu. Faza pierwsza została wydzielona jako część akwizycyjna, w której uzyskujemy zbiór wejściowy obiektów A. Konkretnie mapa Polski z podziałem administracyjnym podlega segmentacji na wiele małych mapek z zachowaniem informacji o pochodzeniu danego obszaru (z którego miejsca dużej mapy). Akwizycja ta dokonywana jest przy pomocy algorytmu Flood_fill[1]. Proces segmentacji może odbywać się wielowątkowo, co znacznie przyśpiesza zakończenie tej fazy.

Kolejnym etapem jest stworzenie reprezentacji odpowiedniej do badania danej cechy. W tym przypadku zostało przyjęte, iż badana jest krawędź oraz pole powierzchni obszaru. Obie te cechy są badane nie wprost, lecz za pomocą pewnych technik, które akceptują szumy w pewnym zakresie. W obu przypadkach posługuję się granulacją [16] jako operatorem, przy pomocy którego powstają ziarna informacyjne. Dla każdego ziarna wyliczam pewne charakterystyki, takie jak ekstrema dla krawędzi, czy też współczynnik pokrycia wnętrza obszaru w ramach danej granuli. Prowadzi to do powstania reprezentacji obiektu dla danej cechy. Reprezentacja ta akceptuje nieścisłości w pewnym zakresie i jednocześnie mierzy odstępstwa jednej reprezentacji obiektu od drugiej. Dzięki temu biorąc odwrotność wartości różnicy pokrycia obiektów możemy określić ich podobieństwo.

[1] en.wikipedia.org/wiki/Flood_fill

Warta podkreślenia jest rola granulacji, jako metody umożliwiającej finalne uzyskanie satysfakcjonujących wyników. Dzięki niej badany komparator nie jest zbyt czuły na drobne różnice pomiędzy cechami obiektów.



Rysunek 5. Diagram aktywności w modelu identyfikacji obszarów geograficznych

Po wyliczeniu danych do reprezentacji przystępujemy do porównywania ze zbiorem obiektów referencyjnych. W tym modelu zostały użyte dwa komparatory. Jeden komparator porównuje reprezentacje krawędziowe, drugi reprezentacje obszarowe. Oba komparatory działają w oparciu o ten sam zbiór referencyjny (lecz różne reprezentacje obiektów). Pierwszy komparator obiektów złożonych używa funkcji przynależności do relacji rozmytej następującej postaci:

$$\mu_{kontur}(a,b) = 1 - DL(a,b) \,/\, \max(n(a), n(b)) \tag{5}$$

gdzie $DL(a,b)$ jest odległością Levenshtein'a [2][10], a $n(a), n(b)$ określają długości ciągów znaków a, b,

drugi zaś:

$$\mu_{pokrycie}(a,b) = 1 - \frac{\sum_i^k |cv_i - cv_i'|}{k} \tag{6}$$

[2] en.wikipedia.org/wiki/Levenshtein_distance

Metoda syntezy	Efektywność	Liczba elementów błędnie zidentyfikowanych
AVG	92,08 %	30
RANK	79,95 %	76
AVGW	91,03 %	34

Tablica 2. Wykaz metod syntezy wyników oraz efektywności działania systemu dla zbioru referencyjnego o 379 elementach

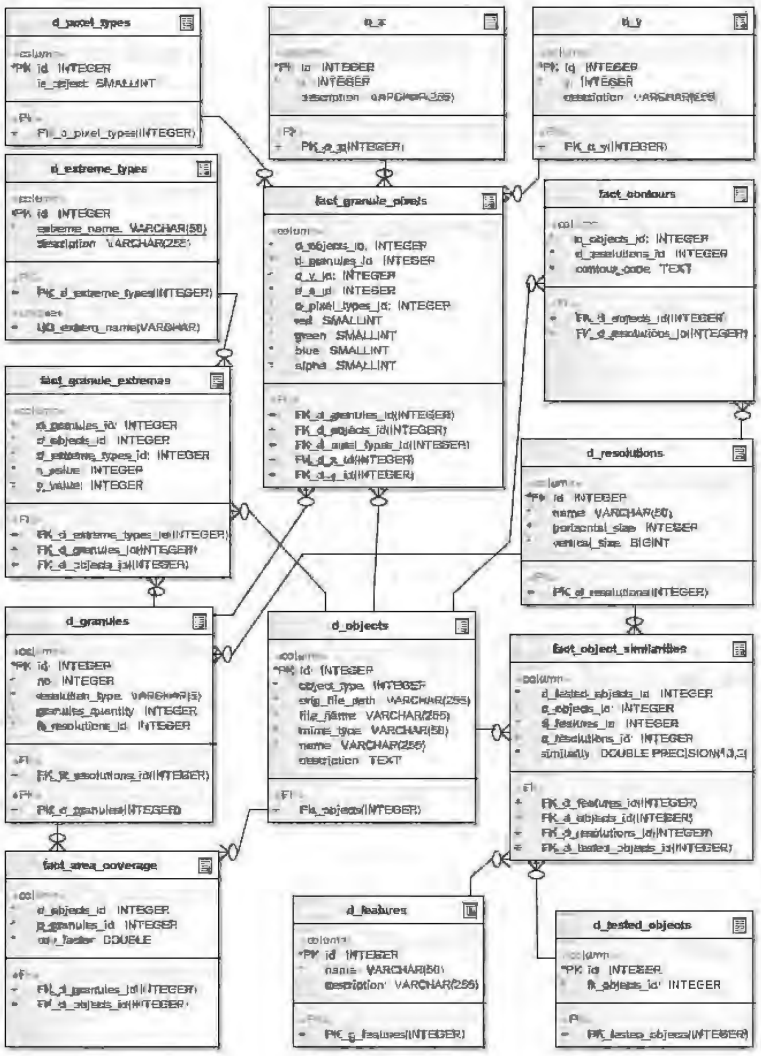
gdzie k>0 jest liczbą granul, cv_i jest numerem i-tej granuli obiektu a, zaś cv'_i jest numerem i granuli obiektu b.

Oba komparatory zwracają wyniki w postaci zbiorów uporządkowanych. Oba mogą działać na pewnym obcięciu tych zbiorów, zatem możliwe do zastosowania są komparatory K_{topN},$K_q$oraz najbardziej typowy K_{st}.

Na koniec następuje połączenie wiedzy przekazanej przez poszczególne jednostki porównawcze i wyliczany jest wynik końcowy stanowiący decyzję modelu. W tym przykładzie, użyta została metoda uśredniania rankingów w oparciu o ten sam zbiór referencyjny. Operacja polega na tym, iż dla każdego obiektu wejściowego wyliczamy średnie arytmetyczne z wyników porównań z poszczególnymi obiektami referencyjnymi. Jako wynik przyjmujemy ten obiekt referencyjny (lub wiele obiektów), dla którego ta średnia wychodzi najwyższa. Możliwe są również inne metody łączenia wyników. Wybrane z nich opisałem w artykule [26]. W zależności od użycia wspomnianych metod syntezy uzyskujemy różną skuteczność systemu. W modelu identyfikacji obszarów geograficznych uzyskano efektywność ok. 92% dla metody syntezy za pomocą średniej arytmetycznej przy identyfikacji obszarów powiatów oraz miast na prawach powiatu w Polsce. W tabeli nr 2 przedstawiono wykaz zbadanych metod łączenia wyników i efektywność końcową algorytmu w zależności od ich wyboru. Badania dokonano na tym samym zbiorze obiektów referencyjnych oraz obiektów badanych.

Jak pokazuje przytoczony przykład, model wymaga efektywnych metod obliczeniowych oraz elastycznego podejścia do charakterystyk. Jak wspomniałem wcześniej, zdecydowałem się na oparcie implementacji modelu na relacyjnych bazach danych w zastosowaniach do hurtowni danych. Dzięki temu operuję na strukturach zwanych kostkami ROLAP w przetwarzaniu z ustalonymi charakterystykami (przy pomocy wymiarów i miar). Jednakże mam w systemie również dane atomowe obiektu (cały obiekt złożony), z którego w każdym momencie mogę wygenerować dodatkowe potrzebne wyniki (charakterystyki, wartości miar, etc). Innym atutem jest możliwość zadawania zapytań na dowolnym podzbiorze danych obiektu złożonego i jednocześnie połączenie z już przeliczonymi agregatami składowanymi w kostkach. Na potrzeby przykładowego modelu stworzonych zostało kilka kostek, które mają różny poziom ziarnistości danych, jednakże kostki połączone są wspólnymi wymiarami (tworząc konstelacje), podlegają tej samej semantyce. Cecha ta jest niezwykle ważna ze względu na możliwość łączenia wyników analiz z poszczególnych kostek. Na rysunku nr 6 został przedstawiony diagram encji stanowiących wymiary oraz tablice faktów dla używanych kostek.

Encje z prefiksem “d_” oznaczają wymiary kostek a z prefiksem “fact_” - docelowe tabele faktów (po konwersji modelu konceptualnego w model fizyczny). Jak widać kostki



Rysunek 6. Schemat ERD kostek ROLAP implementacji modelu identyfikacji map konturowych

oparte są na schemacie gwiazdy, który zapewnia efektywność przetwarzania danych poprzez minimalizację złączeń w zapytaniach (kosztem redundancji danych szeroko stosowanej w hurtowniach danych).

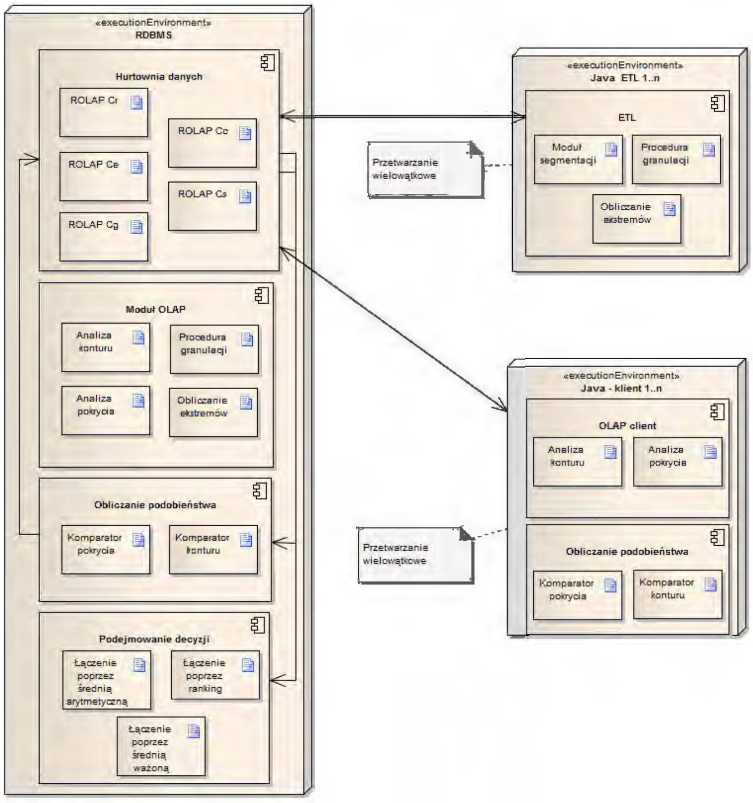
Implementację modelu możemy podzielić na dwie zasadnicze warstwy: serwerową implementowaną po stronie RDBMS oraz kliencką (w moim zastosowaniu aplikacja w języku Java). Dodatkowo warstwę pośrednią stanowią narzędzia klasy ETL, za pomocą których wykonane mogą zostać zadania obu warstw. Na rysunku nr 7 przedstawiony został diagram wdrożenia dla opisywanego modelu. Łatwo zauważyć, iż pewne funkcjonalności są redundantne. Wynika to z faktu, iż mogą one być wykonane w różnych warstwach, a dopiero konkretny postawiony problem, rozstrzyga, w którym miejscu najefektywniej można dokonać danej operacji. Dzięki takiej implementacji, zredukowane zostały ewentualne ograniczenia, które mogą być narzucane przez poszczególne technologie (np. RDBMS i języki 4 poziomu, typu 4GL, PLSQL, etc. lub odwrotnie - języki programowania używane do implementacji części klienckiej nie optymalizowane stricte do przetwarzania dużych wolumenów danych). W takiej architekturze, w zależności od badanego problemu, wybierzemy odpowiednią warstwę dla poszczególnych etapów przetwarzania i analizy danych.

Z punktu widzenia warstwy serwerowej jest to typowe podejście scentralizowane, gdzie dane przechowywane są w jednym środowisku RDBMS, gdzie możemy uzyskać dostęp do poszczególnych etapów pośrednich. Warstwa kliencka natomiast reprezentuje tutaj model przetwarzania rozproszonego, współbieżnego, gdzie wiedza na temat kontekstu obliczeń jest mocno ograniczona. Podejście rozproszone wykorzystuje pewną własność komparatorów obiektów złożonych przytoczoną w rozdziale 2.1. Własność ta zapewnia możliwość podziału zadania na części w ramach stworzenia reprezentacji obiektów, jak również wykonania porównań poprzez podział na rozłączne zbiory zarówno zbioru wejściowego jak i referencyjnego. Uzyskane wyniki dla poszczególnych podzbiorów należy wtedy połączyć i uporządkować przy pomocy funkcji charakterystycznej danej dla konkretnego komparatora obiektów złożonych.

Jak wynika z diagramu na rysunku nr 7, warstwy zostały podzielone na pewne bloki grupujące pewne funkcjonalności, np: “Hurtownia danych”, ”Moduł OLAP”, “Wyliczanie podobieństwa”, “Podejmowanie decyzji”. Poszczególne bloki mogą być producentami i konsumentami danych. Dlatego też implementacja opiera się na klasycznym algorytmie producentów i konsumentów, czyli synchronizacji operacji. Pewne funkcjonalności nie mogą być wykonane, jeśli inny blok nie wyprodukuje danych (np. decyzja nie może być podjęta, jeśli blok wyliczania podobieństwa nie dostarczył jeszcze informacji).

4 Kierunki dalszych badań

Praca pozostawia wiele otwartych zagadnień, które mogą być tematem dalszych badań nad komparatorami. Przede wszystkim należy wspomnieć o metodach uczenia [4], które do zaprezentowanego modelu mogą być zastosowane w kilku miejscach w odniesieniu do różnych danych. Pierwszym elementem jest zbiór referencyjny, którego zawartość jest decydująca w procesie podejmowania decyzji przy użyciu wspomnianego modelu. Dlatego też dobór nowych elementów zbioru będzie wpływał na uzyskiwane wyniki. Proces uczenia zatem będzie tu potrzebny ze względu na umożliwienie adaptacji modelu do



Rysunek 7. Architektura systemu implementującego proponowany model identyfikacji map konturowych

nowych przypadków, jednakże ewentualne błędy mogą pociągać za sobą ważne konsekwencje. Innym elementem może być dobór wartości parametru aktywacji rozwiązania związanego z elementami zbioru referencyjnego. Tutaj również możliwe jest zastosowanie metod optymalizacji doboru parametrów w taki sposób, aby zapewnić dobre wyniki rozwiązań minimalizując jednocześnie koszty obliczeniowe.

Inny aspekt to efektywna reprezentacja obiektów oraz szybkie ich przetwarzanie. Jak wspomniano na wstępie, w pracy przyjęta jest jedna z wielu możliwych dróg, począwszy od reprezentacji, agregacji, kończąc na całościowym schemacie przetwarzania. Zaproponowana metodyka nie narzuca tych kwestii, zakłada jedynie iż będą efektywne, ponieważ porównywanie dużych zbiorów jest kosztowne. Zatem możliwe jest zbadanie innych reprezentacji i ich wpływu na skalowalność rozwiązania.

Kolejnym aspektem możliwym do zbadania jest budowa struktur warstwowych w oparciu o komparatory (analogicznie do sieci neuronowych). Podejście to mogło by być stosowane dla przypadku dekompozycji obiektu złożonego i badania podobieństwa względem zupełnie różnych zbiorów referencyjnych.

Ostatnim aczkolwiek niezwykle istotnym kierunkiem dalszych badań jest adaptacja metod uczących do poszczególnych mechanizmów modelu (zbiór reguł wyjątków, hierarchia zbiorów referencyjnych, skład zbiorów referencyjnych). Można w przyszłości wykorzystać schematy aproksymacyjne [22], za pomocą których budowane mogą być klasyfikatory odpowiedzialne za skład zbiorów referencyjnych oraz hierarchii tych zbiorów.

5 Podsumowanie

Przedstawiona w pracy metodyka budowy modelu oraz sam model są pewnego rodzaju dopełnieniem dla innych metod porównywania obiektów w różnych zastosowaniach. W pracy została pokazana, możliwość konstrukcji struktur porównujących obiekty oraz skuteczne ich użycie do w pełni strukturalizowanych danych wejściowych (obiektów złożonych) w celu uzyskania satysfakcjonujących wyników. Zróżnicowane zastosowania opisanego modelu dowodzą łatwej implementacji w różnych środowiskach i systemach.

Praca zostawia pewne otwarte kwestie teoretyczne i projektowe, które mogą być celem dalszych badań naukowych. Kierunki te zostały omówione w osobnym rozdziale.

Zostało również pokazane praktyczne zastosowanie, za pomocą którego zrealizowano postawione cele z wysoką skutecznością 92% (identyfikacja obszarów geograficznych). We wcześniejszych publikacjach był opisywany model działający na obiektach złożonych typu teksty. Opisany tam model ma praktyczne zastosowanie w module standaryzacji danych (dopasowywania danych wejściowcyh do słownika referencyjnego) oraz do moderacji forum internetowego poprzez dopasowanie występujących fraz we wpisywanym tekście do słownika referencyjnego fraz zakazanych.

Literatura

1. Arabas J.: Wykłady z algorytmów ewolucyjnych, WNT (2000)
2. Boehm C., Berchtold S., Keim D.: Searching in High-Dimensional Spaces—Index Structures for Improving the Performance of Multimedia Databases, ACM Computing Surveys, Vol. 33, No. 3, September 2001, pp. 322–373.
3. Butcher S., Clarke C., Cormack G.: Information Retrieval, Implementing and Evaluating Search engines, MIT Press (2010)
4. Cichosz P.: Systemy uczące się, WNT (2007)
5. Cytowski J., Gielecki J., Gola A.: Cyfrowe przetwarzanie obrazów medycznych, Algorytmy. Technologie. Zastosowania, Exit (2008)
6. Deb S. (ed.): Multimedia Systems and Content-Based Image Retrieval, IGI Global (2004)
7. Garcia-Molina H., Ullman J., Widom J.: Database Systems: The Complete Book. Prentice-Hall (2008)
8. Horstmann C., Cornell G.: Java 2, techniki zaawansowane, Helion (2005)
9. Kacprzyk J.: Multistage Fuzzy Control: A Model-Based Approach to Fuzzy Control and Decision Making. Wiley (1997)
10. Levenshtein V.: Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady 10 :707–710 (1966)
11. Luckham D.: The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems, Addison-Wesley (2002).

12. Malmstadt H. V., Enke C. G., Crouch S. R.: Electronics and instrumentation for scientists, Benjamin/Cummings Pub. Co., (1981)
13. M. Nesenbergs, V. O. Mowery: Logic Synthesis of Some High-Speed Digital Comparators, Bell System Technical Journal, v38: i1 (1959)
14. Osowski S.: Sieci neuronowe do przetwarzania informacji, Oficyna Wydawnicza Politechniki Warszawskiej (2000)
15. Pal S. K., Shiu S. C. K.: Foundations of soft case-based reasoning, Willey-interscience (2004)
16. Pedrycz W., Kreinovich V., Skowron A. (Eds.): Handbook of Granular Computing. Wiley (2008)
17. Pękalska E., Duin R.: The Dissimilarity representation for pattern recognition, World Scientific (2005)
18. Reed T.: Digital Image Sequence Processing, Compression and Analysis, CRC Press (2005)
19. Russ, J.: The Image Processing Handbook (the 5th Edition). CRC Press (2007)
20. Rutkowski L.: Metody i techniki sztucznej inteligencji, PWN (2006)
21. Shannon C. E.: A Mathematical Theory of Communication, The Bell System Technical Journal, Vol. 27, pp. 379–423, 623–656, (1948)
22. Skowron A., Szczuka M.: Approximate Reasoning Schemas: Classifier for Computing With Words
23. Skowron A., Stepaniuk J., Peters J. F.: Towards Discovery of Relevant Patterns from Parametrized Schemas od Information Granule Construction
24. Sosnowski Ł.: Inteligentne dopasowanie danych przy użyciu teorii zbiorów rozmytych w systemach przetwarzania danych, Analiza systemowa w finansach i zarządzaniu T.11 pod redakcją prof. J. Hołubca (2009)
25. Sosnowski Ł.: Budowa systemu porównywania obiektów złożonych, Analiza systemowa w finansach i zarządzaniu T.12 pod redakcją prof. J. Hołubca (2010)
26. Sosnowski Ł.: Identification with compound object comparators - technical aspects, Techniki informacyjne teoria i zastosowania, T.1 pod redakcją prof. J. Hołubca (2011)
27. Sosnowski Ł., Ślęzak D.: Comparators for Compound Object Identification. In: Proc. of RSFDGrC, LNAI 6743, pp. 342-349, Springer (2011)
28. Sosnowski Ł., Ślęzak D.: RDBMS Framework for Contour Identification. In: Proc. of the international workshop CS&P 2011, Białystok University of Technology pp. 487-498 (2011)
29. Stąpor K.: Automatyczna klasyfikacja obiektów, EXIT (2005)
30. Ślęzak D., Sosnowski Ł.: SQL-Based Compound Object Comparators: A Case Study of Images Stored in ICE. In: Proc. of ASEA, CCIS 117, pp. 304-317, Springer (2010)
31. Ślęzak D.: Compound Analytics of Compound Data within RDBMS Framework – Infobright's Perspective. In: Proc. of FGIT. LNCS, vol. 6485, pp. 39–40. Springer, Heidelberg (2010)
32. Ślęzak D., Eastwood V.: Data Warehouse Technology by Infobright. In: Proc. of SIGMOD, pp. 841–845. ACM, New York (2009)
33. Ślęzak D., Szczuka M.: Rough Neural Networks for Complex Concepts, RSFDGrC 2007, LNAI 4482, pp. 574–582, (2007)
34. Ślęzak D., Wróblewski J., Eastwood V., Synak P.: Brighthouse: An Analytic Data Warehouse for Ad-hoc Queries. PVLDB 1(2): 1337-1345 (2008)
35. Szczepaniak P.: Obliczenia inteligentne, szybkie przekształcenia i klasyfikatory, EXIT (2004)
36. Todman C., Projektowanie hurtowni danych, WNT (2003)
37. Wrycza S., Marcinkowski B., Wyrzykowski K.: Język UML 2.0 w modelowaniu systemów informatycznych, Helion (2006)
38. Zadeh L.: Fuzzy Sets, Information and Control, vol. 8, pp. 338-353 (1965)

Compound objects identifications with comparators

Abstract.. The article presents theoretical foundations and practical implementation of the compound object identification methodology based on information granules, fuzzy relations, and the architecture of comparators. You can find practical commercial examples in this paper.

ISBN 9788389475442