STUDIA INFORMATICA 2011
Volume 32 Number 2A (96)

Agnieszka NOWAK-BRZEZIŃSKA, Tomasz JACH
Uniwersytet Śląski, Instytut Informatyki

WNIOSKOWANIE W SYSTEMACH Z WIEDZĄ NIEPEŁNĄ

Streszczenie. Autorzy niniejszego artykułu prezentują nowe podejście do problemu wiedzy niepełnej w systemach wspomagania decyzji. W tym celu stosowane są metody analizy skupień, służące do grupowania reguł w systemie. Artykuł ten przedstawia wyniki badań na temat wpływu parametrów algorytmu grupowania Agnes na jakość grup.

Słowa kluczowe: analiza skupień, grupowanie, systemy wspomagania decyzji, wiedza niepełna

INFERENCE PROCESSES IN DECISION SUPORT SYSTEMS WITH INCOMPLETE KNOWLEDGE

Summary. Authors propose new approach to vagueness problem in decision support systems. To achieve optimal solutions by clustering decision rules, cluster analysis methods are being used. This paper states the results of experiments regarding the influence of Agnes' algorithm to the quality of clustering process.

Keywords: cluster analysis, clustering, decision support systems, uncertain knowledge

1. Wprowadzenie

Systemy wspomagania decyzji stanowią ważną część dzisiejszej informatyki. Przy obecnym, szybkim napływie informacji ich przetwarzanie oraz interpretacja coraz częściej muszą zostać powierzone automatom. Jeszcze do niedawna to ekspert–człowiek dokonywał oceny zastanych warunków i na podstawie swojego doświadczenia podejmował odpowiednią decyzję. Dziś, spora część z tych decyzji została powierzona komputerom, które spełniają swoje zadanie w coraz lepszy sposób.

Jak powszechnie wiadomo, zarówno ekspert–człowiek, jak i odpowiedni algorytm najlepiej wykonują swoją pracę w przypadku, gdy ich wiedza na temat rozpatrywanego zjawiska jest pełna, niesprzeczna i kompletna. Nierzadko można się niestety spotkać z sytuacją, gdy tak komfortowy przypadek jest niemożliwy do otrzymania. Coraz większa niepewność wiedzy (rozumiana poprzez sprzeczne, zapisane obserwacje) oraz jej niepełność (rozumiana jako brak pewnych obserwacji) powoduje spadek jakości rozwiązań z dziedziny systemów wspomagania decyzji.

Stwarza to konieczność poszukiwania nowych metod na radzenie sobie z takimi problemami. Autorzy skupili się w tym artykule na metodach wykorzystujących analizę skupień w celu grupowania reguł w SWD (Systemach Wspomagania Decyzji) [1] celem poprawy ich jakości w sytuacji niepełności i niepewności wiedzy.

1.1. Przegląd dostępnych rozwiązań

Autorzy stosują nowatorskie podejście (między innymi zastosowanie analizy skupień w SWD) w celu umożliwienia przeprowadzenia wnioskowania w systemach z wiedzą niepełną. Dotychczasowe rozwiązania opierały się głównie na wyliczaniu bądź też uzupełnianiu brakujących danych tak, aby osiągnąć stan, w którym nie będzie już niepełności wiedzy. Takie podejście wywodzi się z klasycznych metod matematycznych i zostało wykorzystane w wielu systemach [2]. Kolejną metodą było wykorzystanie teorii zbiorów przybliżonych, w myśl której konstruowane są dolne i górne przybliżenia brakujących wartości [3]. W obecnej chwili teoria ta jest dalej rozwijana [4]. Odmienne podejście proponują autorzy w publikacji [5]. W tym przypadku, do wnioskowania w systemach z wiedzą niepełną używa się teorii zbiorów rozmytych. Do rozwiązania postawionego na wstępie problemu można również użyć sieci Bayesa, co czynią autorzy [6]. Reprezentacja wiedzy jest tutaj oparta na klasycznym rachunku prawdopodobieństwa. Kolejne z podejść zakłada korzystanie z sieci neuronowych [7]. Autorzy starają się udowodnić, iż hybrydowe metody, łączące wiedzę teoretyczną oraz zbiór zaklasyfikowanych przykładów w połączeniu ze zbudowaniem sztucznej sieci neuronowej, są w stanie skutecznie radzić sobie ze skutkami niepełności wiedzy.

1.2. Struktura bazy danych

Tabela 1

Parametry baz danych

	Baza Wine	**Baza Abalone**
Liczba atrybutów	14	8
Liczba obiektów	178	4177
Wygenerowana liczba reguł	115	3079
Rodzaj atrybutów	Numeryczne	Kategoryczne, numeryczne
Puste wartości	brak	brak

Dane, na których operuje proponowany system, powstały poprzez wygenerowanie reguł minimalnych systemem RSES [8] z ogólnodostępnych baz danych *Machine Learning Repository* [9]. Do eksperymentów wybrano bazy Wine oraz Abalone. Podstawowe parametry tych baz przedstawia tabela 1.

Każda z reguł ma postać:

```
(attr4=8600)&(attr8=177)=>(class=2)
(attr4=8600)&(attr1=151)=>(class=2)
(attr4=8600)&(attr7=30)=>(class=2)
```

Część konkluzyjna od części przesłankowej oddzielona jest operatorem wynikania (=>), przesłanki połączone są spójnikiem logicznym „&”.

1.3. Systemy wspomagania decyzji

Klasyczny system wspomagania decyzji rozumiany jest przez autorów jako kombinacja bazy wiedzy oraz algorytmów wnioskowania. Baza wiedzy jest dana w postaci zbioru reguł. Każda reguła składa się z dwóch części: przesłankowej oraz konkluzyjnej. Przesłanki (będące deskryptorami, czyli parami atrybut-wartość) połączone są spójnikiem logicznym „AND”. Klasyczny system wspomagania decyzji oferuje możliwość wyprowadzania nowej wiedzy z sytemu na podstawie znanych wcześniej faktów. Fakty te umieszczone są w osobnym zbiorze i są również zapisane w postaci deskryptorów. Klasyczne wnioskowanie w przód uaktywnia te reguły, których wszystkie przesłanki mają swoje odzwierciedlenie w zbiorze faktów. Po uaktywnieniu reguły, jej część konkluzyjna zostaje dowiedziona i umieszczona w bazie faktów jako nowe informacje wyprowadzone z systemu [1].

Istotną wadą klasycznego SWD jest możliwość impasu – czyli braku możliwych reguł do uaktywnienia. Dzieje się tak wtedy, gdy zbiór faktów nie zawiera dostatecznej liczby informacji, pozwalającej na uaktywnienie którejkolwiek z reguł. Proponowany system radzi sobie z tym problemem w taki sposób, iż przedstawi użytkownikowi reguły, których przesłanki są albo w największym stopniu pokryte w zbiorze faktów[1], albo też – wartości deskryptorów w przesłankach tych reguł są najbliższe tym, które znajdują się w zbiorze faktów. Uaktywnienie tych niepewnych reguł będzie mogło mieć miejsce po akceptacji przez użytkownika niepewności tak wygenerowanej wiedzy.

Kolejną zaletą przedstawianego systemu jest łatwiejszy sposób panowania nad lawinowo rosnąca liczbą nowych faktów i reguł, uaktywnianych w klasycznym wnioskowaniu w przód. Proponowany system, pomimo stosunkowo licznego zbioru reguł, dzięki zaimplementowanym mechanizmom grupowania, pozwala na bardzo szybkie odnalezienie interesującej reguły

[1] Termin *pokrycie* jest rozumiany przez autorów jako istnienie deskryptorów zgodnych z tymi w regule w zbiorze faktów. Innymi słowy: reguła będzie pokryta zbiorem faktów w maksymalnym stopniu wtedy i tylko wtedy, gdy wszystkie przesłanki reguły będą również należeć do zbioru faktów.

na podstawie przesłanek lub też konkluzji, które dana reguła zawiera. Dzięki temu wydajność zostaje znacznie zwiększona.

2. Grupowanie reguł w SWD

Jednym z głównych zadań dziedziny, jaką jest analiza skupień jest podział zestawu danych na sensowne grupy (skupienia). Ma to miejsce tylko za pomocą wiedzy otrzymanej z danych wejściowych: jej analizy z wykorzystaniem metod statystycznych, matematycznych oraz innych. To wzajemne powiązania pomiędzy obiektami oraz ich relacje (często niewidoczne na pierwszy rzut oka) umożliwiają przeprowadzenie procesu grupowania. Istotnym założeniem jest zachowanie zależności mówiącej o maksymalizowaniu wzajemnego podobieństwa obiektów wchodzących w skład danej grupy oraz minimalizacji podobieństwa poszczególnych grup jako całości.

Jedną z metod analizy skupień są tzw. aglomeracyjne algorytmy hierarchiczne, których przedstawicielem jest omawiany algorytm Agnes. Ogólną zasadą działania zarówno tego algorytmu, jak i pozostałych z tej grupy jest początkowe założenie, iż każda z reguł jest reprezentantem oddzielnej grupy. W kolejnych krokach algorytm dąży do odnalezienia najbardziej podobnych reguł, a następnie – łączy je w większe skupienia [10]. Autorzy używają algorytmu Agnes (*Aglomerative NESting*) [11] do grupowania reguł wchodzących w skład bazy danych. Reguły są otrzymywane ze zbiorów dostępnych w Machine Learning Repository [9] poprzez ich import do programu RSES i wygenerowanie reguł minimalnych algorytmem LEM2 [2].

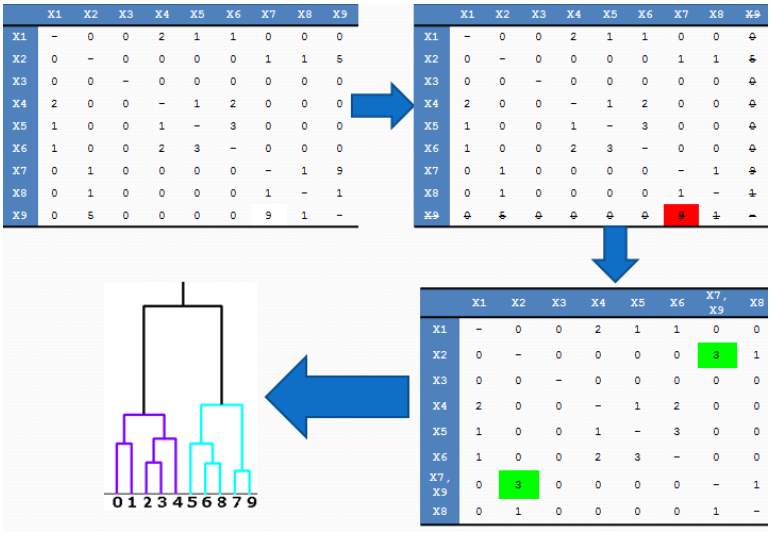
2.1. Wybór algorytmu grupowania

Agnes, algorytm użyty do grupowania reguł, wywodzi się z grupy algorytmów hierarchicznych. Jego działanie określa schematycznie rys. 1.

Pierwowzorem do stworzenia przedstawionego systemu był m.in. system SMART Saltona. Ideą jego działania było grupowanie dokumentów na podstawie ich podobieństwa [12]. Podobnie jak w systemie proponowanym przez autorów, wyszukiwanie odbywało się w znacznie krótszym czasie w stosunku do przeszukiwania liniowego, ze względu na wykorzystanie reprezentantów grup. Pytanie trafiające do systemu zostaje porównane z reprezentantami wyliczonymi w trakcie procesu grupowania. Dzięki temu, następuje znaczne zmniejszenie liczby porównań skutkujących znalezieniem reguł relewantnych. Podobna idea przyświecała także twórcom systemu Carrot2 [13].

Elementem, który najbardziej wpływa na jakość ostatecznego zgrupowania jest sposób tworzenia macierzy podobieństwa, a jedyną zauważalną wadą algorytmu jest jego relatywnie

wysoka złożoność obliczeniowa [rzędu $O(n^2)$]. Do zalet z całą pewnością należy zaliczyć odporność na obiekty odległe (skrajnie niepodobne w stosunku do wszystkich innych), a także fakt, że przeszukiwanie drzewa odbywa się w czasie logarytmicznym – złożoność tego procesu to O(log n), typowa dla drzew binarnych (jakie tworzy wspomniany algorytm).



Rys. 1. Schemat działania algorytmu Agnes
Fig. 1. Conception of Agnes algorithm

W pierwszym kroku algorytmu następuje generowanie kwadratowej macierzy podobieństwa, której rozmiar jest równy liczbie reguł zapisanych w systemie. Algorytm na przecięciu i-tego wiersza i j-tej kolumny wylicza podobieństwo dwóch reguł do siebie. Początkowo autorzy zaproponowali prosty sposób wyliczania podobieństwa dwóch reguł:

$$simpleSimilarity = \overline{\overline{d_p \cap d_q}} \tag{1}$$

Co najmniej równie efektywnym podejściem jest obliczanie ważonego współczynnika podobieństwa:

$$weightedSimilarity = \frac{\overline{\overline{d_p \cap d_q}}}{\overline{\overline{d_p \cup d_q}}} \tag{2}$$

W obydwu przypadkach d_p oraz d_q oznaczają zbiory deskryptorów reguły p oraz q.

Po wyliczeniu macierzy podobieństwa, następuje właściwy algorytm grupowania. Wyszukiwana jest maksymalna wartość współczynnika podobieństwa określająca, które reguły połączyć. W podanym przykładzie są to reguły X7 oraz X9, które zostają połączone w jedno skupienie. Następnym krokiem będzie wyliczenie wartości podobieństwa nowo powstałego skupienia do pozostałych reguł. Tutaj również algorytm pozwala na manipulacje sposobami jego wyliczania. W przedstawionym systemie zaimplementowano trzy kryteria łączenia skupień [14, 15]:

1. Pojedyncze wiązanie (ang. *Single Linkage*) – używana jest funkcja *minimum* (wybierająca minimalną wartość spośród argumentów), która bierze pod uwagę odległość do najbliż-

szego sąsiada. Metoda pojedynczego wiązania ma tendencje do tworzenia małej liczby heterogenicznych grup.

$$d_{(p+q),i} = \min(d_{pi}, d_{qi})$$

2. Metoda uśrednionego wiązania (ang. *Average Linkage*) – w metodzie tej odległość między dwoma skupieniami obliczana jest jako średnia odległość między wszystkimi parami obiektów należących do dwóch różnych skupień. Metoda ta jest efektywna, gdy obiekty formują naturalnie oddzielone grupy.

$$d_{(p+q),i} = \text{avg}\,(d_{pi}, d_{qi})$$

3. Całkowite wiązanie (ang. *Complete Linkage*) – stosujące funkcję *maximum*, która spośród argumentów wybiera maksymalną wartość. Dzięki temu, zbudowane grupy są bardziej jednorodne, albowiem funkcja *maximum* bierze pod uwagę maksymalną odległość pomiędzy dwoma obiektami należącymi do grupy. Z tego też powodu, metoda ta jest również zwana „metodą najdalszego sąsiada".

$$d_{(p+q),i} = \max(d_{pi}, d_{qi})$$

gdzie:

d_{pi} – odległość (niepodobieństwo) pomiędzy grupami p-tą a i-tą,

d_{qi} – odległość (niepodobieństwo) pomiędzy grupami q-tą a i-tą,

$d_{(p+q),i}$ – odległość (dystans) pomiędzy grupą powstałą z połączenia grup p oraz q oraz grupą i-tą.

Po zakończeniu grupowania (liczba kroków będzie równa liczbie reguł w systemie) otrzymujemy drzewo zwane *dendrogramem*. Algorytm w ostatnim stadium dokonuje przycięcia takiego drzewa, w celu otrzymania pożądanej liczby grup.

Jak widać, algorytm dokonuje pełnego procesu klasteryzacji, buduje pełne drzewo aż do momentu, w którym wszystkie reguły znajdą się w jednym skupieniu. Dzięki temu, użytkownik może dynamicznie wybierać liczbę grup (ściślej: wysokość, na której zostaje przycięty dendrogram) bez powtarzania procesu grupowania. Cecha ta jest bardzo przydatna ze względu na brak możliwości ścisłej i dokładnej oceny liczby grup występujących w zbiorze danych.

Po wykonaniu przedstawionego algorytmu, wygenerowane zostają skupienia złożone z reguł. Następnie, wyznaczany jest reprezentant każdej grupy. Na tym etapie prac nad systemem, autorzy ograniczyli się do dwóch sposobów jego wyznaczania.

Pierwszym ze sposobów jest dokonanie sumy logicznej wszystkich deskryptorów wchodzących w skład reguł tworzących grupy. Powoduje to, że reprezentant staje się stosunkowo długi, lecz pamiętajmy o tym, że każda z grup zawiera tylko reguły charakteryzujące się wysokim podobieństwem do siebie. Fakt ten powoduje, iż długość reprezentanta wyznaczanego jako wyrażenie złożone z deskryptorów, połączonych operatorem logicznym „OR" i tak jest znacznie krótsza niż po prostu zestaw wszystkich cech, opisujących reguły w grupie (inaczej

mówiąc: jeśli cechy te się powtarzają, reprezentant składa się tylko z jednego powtórzenia).

Drugi sposób wykorzystuje operator „AND", tworząc reprezentanta jako iloczyn cech wspólnych dla wszystkich dokumentów wchodzących w skład grupy. Podejście to znacznie skraca jego długość, lecz niestety tworzy reprezentanta w sposób bardzo restrykcyjny, a co za tym idzie – liczba cech opisujących taką grupę jest niewielka. Skutkuje to trudnością w późniejszym rozróżnieniu grup. Przykładowo, mając daną grupę złożoną z dwóch reguł:

```
Reg. 1:  (attr7=26)&(attr4=9500)=>(class=1)
Reg. 2: (attr7=26)&(attr4=10000)=>(class=1)
```

reprezentant utworzony za pomocą operatora „OR" będzie mieć postać:

```
(attr7=26), (attr4=9500), (attr4=10000), (class=1)
```

natomiast reprezentant utworzony jako iloczyn cech wspólnych:

```
(attr7=26), (class=1)
```

Jak widać, autorzy traktują klasę decyzyjną jako dodatkowy deskryptor, zarówno w przypadku liczenia podobieństwa dwóch reguł, jak i wyliczania reprezentanta grupy. Podejście to jest uzasadnione faktem, iż wartość decyzji pamiętana w regule jest kluczowym elementem, nierzadko pozwalającym na poprawne podzielenie reguł w systemie.

Mając dane te wszystkie informacje, można przystąpić do samego procesu wyszukiwania. W obecnej wersji systemu wyszukiwanie reguł relewantnych odbywa się poprzez porównanie zapytania złożonego z iloczynu deskryptorów (par atrybut-wartość) z reprezentantami grup. Wynikiem jest grupa mająca największe podobieństwo swojego reprezentanta do zadanego pytania. Algorytm po zwróceniu wyników podaje parametry efektywności wyszukiwania, omówione w części eksperymentalnej niniejszego artykułu.

3. Eksperymenty obliczeniowe

Przedmiotem eksperymentów jest pomiar parametrów kompletności i dokładności[2] wyszukiwania reguł relewantnych w procesie wnioskowania względem obserwacji początkowych (faktów).

Należy tu zauważyć, że taki system będzie potrafił znajdować zarówno reguły w pełni pokrywające zadane fakty, jak i takie reguły, które tylko w pewnym stopniu pokrywają zbiór obserwacji. Taki stan rzeczy pozwoli na wyprowadzanie nowej wiedzy z systemu, nawet przy

[2] Kompletność rozumiana, jako stosunek liczby odnalezionych reguł relewantnych do liczby wszystkich reguł relewantnych w systemie. Dokładność jest rozumiana jako stosunek liczby reguł relewantnych odnalezionych do liczby wszystkich reguł odnalezionych przez system. Wszystkie porównania w obrębie eksperymentów przeprowadzana są w stosunku do reprezentanta grupy.

niepełnej informacji. Oczywiście wiąże się z tym fakt, że parametry kompletności i dokładności nie będą wtedy przyjmować wartości optymalnych, gdyż w rezultacie nie uda się znaleźć wszystkich reguł do uaktywnienia.

Każdy eksperyment składał się z czterech przypadków testowych:

1. Istnieje przynajmniej jedna taka reguła w bazie wiedzy, której wszystkie przesłanki są prawdziwe (są faktami w bazie wiedzy). Uwzględnia się zatem wśród obserwacji te wszystkie, które występują w części warunkowej przynajmniej jednej reguły.
2. Do początkowo pustego zbioru faktów dopisywane są losowo wybrane deskryptory spośród wszystkich obecnych w systemie.

   Aby sprawdzić skuteczność radzenia sobie systemu z dużą dozą losowości, autorzy wykonali test, w którym system miał odnaleźć najbardziej relewantną grupę w stosunku do całkowicie losowego zbioru deskryptorów, będących pytaniem.
3. Zbiór faktów składa się ze wszystkich prócz jednego deskryptorów pokrywających losowo wybraną regułę w bazie.

   Test ten sprawdza czy system potrafi odnaleźć regułę (oraz inne, najbardziej do niej podobne) w przypadku, gdy jedna z przesłanek nie zostanie uwzględniona w pytaniu. W systemach z wiedzą niepełną umiejętność ta jest szczególnie istotna.
4. Dokładnie jedna para atrybut-wartość stanowi cały zbiór faktów.

   Autorzy sprawdzali również czy system poradzi sobie z pytaniem ogólnym, zawierającym jeden deskryptor użyty w systemie. Spodziewaną odpowiedzią jest dość liczny zbiór reguł.

3.1. Metody wyboru reprezentanta grupy

Tabela 2

Eksperyment – wybór reprezentanta grupy

Nr testu	Kompletność	Dokładność	Liczba reguł	**Reprezentant „AND”**	**Reprezentant „OR”**	Liczba reguł
Baza Wine						
1	1	**0,67**	3	0,6	**1**	3
2	**1**	**0,4**	1	**1**	**0,4**	1
3	**0,5**	0,33	6	0,33	**1**	6
4	**0,5**	**1**	2	0,25	**1**	2
Baza Abalone						
1	**1**	0,75	2	0,32	**1**	6
2	**1**	0,5	165	0,02	**0,7**	505
3	**1**	0,8	2	0,625	**1**	3
4	**1**	**1**	6	0,5	**1**	29

Pierwszym problemem, którego wyniki autorzy chcieliby przedstawić jest problem wyboru reprezentanta stojącego na czele grupy. Obie koncepcje wykorzystane przez autorów zostały omówione w poprzednich częściach tego artykułu. Eksperymenty mają na celu spraw-

dzenie, które z podejść (proste podobieństwo lub ważone podobieństwo) wygeneruje lepsze rezultaty. Tabela 2 przedstawia wyniki dla poszczególnych baz danych, ze względu na sposób liczenia reprezentanta.

Jak widać, reprezentant typu „AND" uzyskiwał znacznie lepsze wyniki kompletności, przy stosunkowo dużych wartościach parametru dokładności. Widać również, iż w tym przypadku grupy były mniej liczne. Pełna kompletność uzyskana została dzięki odnalezieniu wszystkich reguł relewantnych do pytania, przy jednoczesnym umiejscowieniu innych, podobnych, nie w pełni relewantnych reguł w skupieniu (parametr dokładności).

Autorzy przeanalizowali wyniki dla reprezentantów tworzonych jako iloczyn albo suma deskryptorów w danej grupie. Okazuje się, że tego typu prosty reprezentant jest całkowicie niewystarczający do zastosowania. Optymalnym rozwiązaniem byłaby metoda pośrednia, niwelująca słabości reprezentanta „AND" (który ma tendencję do tworzenia mało rozróżnialnych grup o podobnym opisie) oraz reprezentanta „OR" (tworzącego bardzo długie opisy, trudne w dalszym przetwarzaniu). Pewną koncepcją, która zostanie wdrożona przez autorów będzie umieszczanie w reprezentancie unikalnych, w skali całego systemu, deskryptorów reguł wchodzących w skład grupy.

Po przeprowadzeniu eksperymentów, autorzy stwierdzili, iż stosunkowo częstą sytuacją jest taka, gdy kilka grup jest tak samo podobnych do zadanego pytania (sytuacja ma miejsce zwłaszcza w przypadku reprezentanta „AND"). Aby polepszyć wyniki kosztem czasu wyszukiwania, należałoby sprawdzać jak wiele reguł w tych grupach, stanowiących odpowiedź przybliżoną, jest rzeczywiście relewantnych do pytania.

3.2. Wybór miary odległości pomiędzy regułami

Tabela 3

Eksperyment – wybór miary odległości pomiędzy regułami

	Proste podobieństwo			**Podobieństwo ważone**		
Nr testu	Kompletność	Dokładność	Liczba reguł	Kompletność	Dokładność	Liczba reguł
Baza Wine						
1	**1**	**0,67**	3	0,4	**0,67**	2
2	**0,67**	**0,4**	1	**0,67**	**0,4**	1
3	0,04	**1**	31	**0,5**	0,5	6
4	**0,25**	**1**	2	**0,25**	**1**	2
Baza Abalone						
1	0,42	1	4	**1**	**1**	1
2	0,01	0,75	411	**0,67**	0,5	1
3	0,4	1	3	**0,5**	0,5	4
4	0	1	1086	**0**	0	2

Drugą kwestią (interesującą autorów) był wybór miary odległości pomiędzy dwoma regułami. Jak wspomniano wcześniej, autorzy zaimplementowali dwa sposoby liczenia podobień-

stwa pomiędzy dwoma regułami. Wyniki eksperymentów przedstawia tabela 3.

W przypadku pytań ogólnych oraz tych, zawierających losowe deskryptory w systemie, odpowiedzi również były nadspodziewanie dobre, aczkolwiek ich wartości kompletności i dokładności – znacznie mniejsze od pozostałych dwóch (kompletny opis jednej z reguł w zbiorze faktów oraz niepełny opis) przypadków testowych.

Podobieństwo ważone wygenerowało stosunkowo lepsze rezultaty. Jakkolwiek wyniki dokładności były lepsze dla prostego podobieństwa, tak (zwłaszcza dla bazy Abalone) wyniki kompletności dla ważonego podobieństwa deklasowały wyniki dla podobieństwa prostego, przy wystarczająco dobrych wynikach dokładności. Podobnie jak w przypadku eksperymentów dotyczących metody wyznaczania reprezentanta, tak i tutaj mamy do czynienia ze znacznie lepszymi dla przedstawionego systemu rezultatami, dla podobieństwa typu ważonego. Powody tego stanu rzeczy są tożsame dla tych, przytoczonych przy omawianiu wyników poprzedniego eksperymentu.

Metoda prostego podobieństwa zdecydowanie lepiej współgra z reprezentantem typu „OR", a metoda ważonego podobieństwa – z reprezentantem typu „AND". W ogólnym przypadku jednak, ważone podobieństwo pozwala na dużo większą dywersyfikację grup i reguł, a co za tym idzie – lepsze ogólne rezultaty.

3.3. Wpływ kryterium łączenia skupień na jakość grup

Ostatnią kwestią poruszoną w tym artykule będzie wpływ kryterium łączenia skupień na jakość grup. Z racji specyficzności procesu grupowania reguł, dopiero eksperymenty mogą dać odpowiedź co do najlepszego podejścia. Wyniki przedstawia tabela 4.

Tabela 4

Eksperyment: kryterium łączenia skupień

	Single linkage			Average linkage			Complete linkage		
Nr testu	Kompletność	Dokładność	Liczba reguł	Kompletność	Dokładność	Liczba reguł	Kompletność	Dokładność	Liczba reguł
Baza Wine									
1	0,4	**0,67**	2	**0,5**	0,33	2	**0,5**	0,33	2
2	**0,67**	0,4	1	0,09	**0,8**	29	**0,67**	0,4	1
3	**0,5**	0,5	6	0,22	0,67	6	0,07	**1**	18
4	**0,25**	**1**	2	0,02	**1**	29	**0,25**	**1**	2
Baza Abalone									
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
2	**0,67**	**0,5**	1	**0,67**	**0,5**	1	**0,67**	**0,5**	1
3	0,5	0,5	4	0,75	0,75	2	**0,8**	**1**	1
4	0	0	2	0	0	0	**1**	**0,25**	2

W trakcie przeprowadzania eksperymentów dotyczących kryterium wiązania reguł w grupy dla bazy Wine autorzy nie potrafili rozstrzygnąć, która z metod da lepsze rezultaty. Dopiero eksperymenty na znacznie większej bazie Abalone rozstrzygnęły jednoznacznie, iż

to metoda całkowitego wiązania będzie lepsza. Problem ten wymaga dalszych, wnikliwych eksperymentów w celu ostatecznego potwierdzenia tej hipotezy.

Przed rozpoczęciem eksperymentów autorzy zakładali, że najlepsze wyniki otrzymywane będą dla kryterium łączenia skupień, jakim jest pojedyncze wiązanie. Założenie to miało swoje podstawy w samej obserwacji działania pojedynczego wiązania – łańcuchowania reguł wzajemnie podobnych do siebie. Eksperymenty jednak udowodniły, zwłaszcza dla zbioru Abalone, że to metoda całkowitego wiązania da w wyniku lepsze rezultaty. Z racji tej obserwacji można wysnuć wniosek, że reguły tak naprawdę tworzą dużą liczbę mało licznych grup, co autorzy zakładali na początku.

3.4. Wnioski z badań i koncepcje dalszej pracy

System pozwolił na znaczne zmniejszenie liczby przeglądanych reguł. Zysk czasowy jest zwłaszcza widoczny dla dużych baz reguł, gdzie w celu odszukania grupy relewantnej trzeba porównać tylko kilku(nastu) reprezentantów grup zamiast każdą regułę w systemie. W efekcie, proponowane rozwiązanie pozwala ograniczyć liczbę reguł, faktycznie analizowaną w procesie wnioskowania, do kilku procent (3-10%). Średni zysk czasowy w aktualnej wersji systemu jest związany z liczbą utworzonych grup i może być zapisany symbolicznie jako $\frac{k}{n}$, gdzie: k to liczba skupień, a n to liczba reguł w systemie. Jednocześnie zakłada się, że porównanie każdej reguły z pytaniem zajmuje jedną jednostkę czasu. W trakcie wnioskowania, w planowanej wersji systemu zysk czasowy będzie ściśle uzależniony od liczby grup oraz ich liczności. Ogólnie mówiąc, im więcej grup, tym zysk czasowy mniejszy.

W zdecydowanej większości przypadków, co obrazują tabele przedstawiające wyniki eksperymentów, system był w stanie poprawnie odnaleźć regułę na podstawie wszystkich przesłanek ją opisujących (przypadek testowy nr 1) oraz wtedy, gdy podane były tylko trzy z czterech deskryptorów, opisujących obiekt. Oprócz reguł relewantnych zostały również zwrócone reguły bardzo do nich podobne, co jest wielce pożądanym zjawiskiem.

W wyniku licznych eksperymentów, najbardziej obiecujące wyniki otrzymano dla metody całkowitego wiązania, reprezentantów jako iloczynów deskryptorów wspólnych oraz ważonej miary podobieństwa.

Obecnie trwają eksperymenty próbujące oszacować metodę wyboru optymalnej liczby skupień w systemach z grupowaniem reguł. Wstępne wyniki wskazują na stosunkowo dużą liczbę grup, konieczną do otrzymania prawidłowych wyników. Duża liczba mało licznych grup pozwoli na otrzymanie skupień, których wewnętrzna spójność będzie wysoka, a co za tym idzie – uaktywnienie wszystkich reguł w skupieniu pozwoli na minimalizację niepewności wnoszonej tym procesem do systemu wspomagania decyzji.

BIBLIOGRAFIA

1. Nowak-Brzezińska A., Wakulicz-Deja A.: Analiza efektywności wnioskowania w złożonych bazach wiedzy. Systemy Wspomagania Decyzji, 2007.
2. Bazan J., Nguyen H. S., Nguyen S. H., Synak P., Wróblewski J.: Rough set algorithms in classification problems. [in:] Polkowski L., Lin T. Y., Tsumoto S.(eds.): Rough Set Methods and Applications: New Developments in Knowledge Discovery in Information Systems. Heidelberg: Physica-Verlag, 2000, s. 49÷88.
3. Pawlak Z.: Rough set approach to knowledge-based decision suport. European Journal of Operational Research. 16 May 1997, s. 48÷57.
4. Latkowski R.: Wnioskowanie w oparciu o niekompletny opis obiektów (praca magisterska). Warszawa: Wydział Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego, 2001.
5. Zadeh L.A., Kacprzyk J.: Fuzzy logic for the management of uncertainty. New York: John Wiley & Sons, 1992.
6. Geiger D., Heckerman D.: Knowledge representation and inference in similarity networks and Bayesian multinets. Artificial Intelligence. 1996, s. 45÷74.
7. Towell G. G., Shavlika J.W.: Knowledge-based artificial neural networks. Artificial Intelligence. 1994, s. 119÷165.
8. Bazan J. G., Szczuka M. S., Wróblewski J.: A new version of rough set exploration system. [in:] Alpigini J. J. et al.(eds.): Third International Conference on Rough Sets and Current Trends in Computing RSCTC. Malvern, PA: Springer-Verlag, 2002, s. 397÷404.
9. Frank A., Asuncion A.: UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science, 2010.
10. Kaufman L., Rousseeuw P. J.: Finding Groups in Data: An Introduction to Cluster Analysis. Wiley, New York 1990.
11. Jain A. K., Dubes R. C.: Algorithms for clustering data. New Jersey: Prentice Hall, 1988.
12. Salton G.: Automatic Information Organization and Retreival. New York, USA: McGraw-Hill, 1975.
13. Osiński S., Weiss D.: Carrot2: An Open Source Framework for Search Results Clustering, 2004.
14. Kumar V., Tan P. N., Steinbach M.: Introduction to Data Mining. Addison-Wesley, 2006.
15. Myatt G. J.: Making Sense of Data A Practical Guide to Exploratory Data Analysis and Data Mining. New Jersey : John Wiley and Sons, Inc., 2007.
16. Wakulicz-Deja A.: Podstawy systemów wyszukiwania informacji. Analiza metod. Warszawa: Akademicka Oficyna Wydawnicza PLJ, 1995.

Recenzenci: Dr inż. Ewa Płuciennik-Psota
Dr inż. Marek Sikora

Wpłynęło do Redakcji 16 stycznia 2011 r.

Abstract

In this paper authors cope with incompleteness in decision support systems. A new method involving cluster analysis is proposed. In the beginning there is a brief description of the problem, followed by summary of known solutions. After that authors propose completely new approach based on cluster analysis and hierarchical grouping algorithms. The summary of databases used in experiments is shown in Table 1. Authors consider variety of factors involved in grouping rules in decision support systems: the problem of choosing groups' representatives', choosing the correct distance measure and optimizing search results by choosing correct group joining criteria. The results of these experiments are shown in Tables 2-4. The paper is summarized in the end where optimal parameters of the algorithm are presented.

Adresy

Agnieszka NOWAK-BRZEZIŃSKA: Uniwersytet Śląski, Instytut Informatyki, ul. Będzińska 39, 41-200 Sosnowiec, Polska, agnieszka.nowak@us.edu.pl.
Tomasz JACH: Uniwersytet Śląski, Instytut Informatyki, ul. Będzińska 39, 41-200 Sosnowiec, Polska, tomasz.jach@us.edu.pl.