Uniwersytet Śląski
Wydział Informatyki i Nauki o Materiałach
Informatyka

Rozprawa doktorska

Optymalizacja procesów wnioskowania z wiedzą niepełną

mgr Tomasz Jach

Promotor: prof. dr hab. inż. Alicja Wakulicz-Deja
Promotor pomocniczy: dr Agnieszka Nowak-Brzezińska

Katowice, 2013 r.

Rodzicom.

Wyrażam zgodę na udostępnienie mojej pracy doktorskiej dla celów naukowo-badawczych.

Data:

Podpis autora:

Słowa kluczowe: *analiza skupień, wnioskowanie, wiedza niepełna, systemy wspomagania decyzji, AHC, mAHC, grupowanie reguł, współczynniki pewności, CF, hierarchiczna baza wiedzy*

Oświadczenie autora pracy

Świadomy odpowiedzialności prawnej oświadczam, że niniejsza praca doktorska została napisana przez mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Data Podpis autora pracy

Spis treści

Podziękowania

Z całego serca chciałbym podziękować mojemu Promotorowi Profesor Alicji Wakulicz-Deji oraz Promotorowi Pomocniczemu Doktor Agnieszce Nowak-Brzezińskiej za nieocenioną pomoc, wsparcie i dużą dawkę motywacji przy pisaniu tej pracy.

Bez ich nadzoru i pomocy praca ta nigdy nie powstałaby w takim kształcie, jak obecnie. Dzięki pracy w Zakładzie Systemów Informatycznych, udziale w licznych konferencjach naukowych i seminariach, pod ścisłym kierunkiem Pani Profesor i Pani Doktor, mogłem wykształcić warsztat niezbędny do przeprowadzenia i opisania przedstawianych tu badań naukowych.

Dziękuję również za trud włożony w rozładowanie sytuacji stresowych, których nie brakuje przy redagowaniu każdej pracy naukowej.

Składam również podziękowania moim najbliższym za ich wyrozumiałość, gdy tego potrzebowałem.

Rozdział 1

Wprowadzenie

Tematem przedstawionej rozprawy są teoretyczne i praktyczne aspekty efektywności Systemów Wspomagania decyzji (SWD) nazywanych często systemami ekspertowymi (SE) i bazujących na tzw. regułowych bazach wiedzy. Systemy te działają w oparciu o procedury rozumowania (wzorowane na rozumowaniu realizowanym przez człowieka) i przeprowadzają proces zwany wnioskowaniem. W literaturze przedmiotu proces wnioskowania wykorzystuje wiedzę dziedzinową, najczęściej przedstawioną w formie łańcuchów przyczynowo-skutkowych typu "Jeżeli warunek, to decyzja" (reguł). Dodatkowo zakłada się uaktywnienie w procesie wnioskowania tylko tych reguł, których części warunkowe (zwane przesłankami) są prawdziwe (są potwierdzone faktami znanymi w systemie). Konkluzje uaktywnionych reguł stają się nowymi faktami uznanymi za prawdziwe. W ten sposób system gromadzi nową wiedzę wspomagającą podejmowanie decyzji.

Celem prowadzonych badań jest propozycja zamiany dotychczasowej struktury bazy wiedzy (w której reguły nie były uporządkowane i nie tworzyły żadnych złożonych struktur) na strukturę skupień reguł podobnych do siebie, zarówno ze względu na przesłanki reguł, jak i konkluzje. Zmiana ta pozwoli optymalizować efektywność wnioskowania w tego typu systemach dzięki szybkiemu znalezieniu reguł do uaktywnienia oraz rozszerzeniu dotychczasowej bazy faktów o nową wiedzę wydobytą poprzez uaktywnienie reguł, których nie wszystkie przesłanki są spełnione. Bardzo często w typowych rzeczywistych systemach wspomagania decyzji mamy do czynienia z sytuacją, gdy nie mamy pełnej wiedzy (bądź pełnego przekonania), która jest niezbędna by podjąć kolejne kroki (decyzje). Opracowanie metody określania stopnia niepełności wiedzy reprezentowanej przez daną regułę

stało się jednym z celów niniejszej rozprawy. Struktura skupień reguł wraz z reprezentantami skupień - będąc nową formą reprezentacji regułowej bazy wiedzy - pozwoli szybko znaleźć regułę bądź skupienie do uaktywnienia w procesie wnioskowania nawet w sytuacji gdy nie mamy pełnej wiedzy o badanej dziedzinie. Dzieje się tak wtedy, gdy zbiór faktów nie zawiera dostatecznej liczby informacji pozwalającej na uaktywnienie którejkolwiek z reguł. W literaturze znaleźć można prace, które również proponowały zmianę struktury bazy wiedzy wykorzystując analizę skupień jak i prace dotyczące reprezentacji niepewności wiedzy. Rozwiązania proponowane w rozprawie znacząco się jednak od tamtych różnią. Ich zaletą jest intuicyjność i łatwość implementacji. Mając zadany zbiór faktów, przegląd bazy wiedzy polega na porównaniu wektora faktów z reprezentantami skupień i wyborze skupienia najbardziej podobnego. Proces takiego przeglądu kończy się w momencie znalezienia skupienia bądź reguły, która pokryje możliwie najpełniej zadany zbiór faktów. Jeśli wyszukana w ten sposób zostanie reguła, której wszystkie przesłanki są prawdziwe, w wyniku procesu wnioskowania jej konkluzja zostanie dopisana jako nowy fakt do bazy wiedzy ze stopniem niepełności równym 1. W przypadku gdy, w wyniku takiego przeszukiwania, znaleziona będzie reguła, która tylko w pewnym stopniu (możliwie największym) pokrywa zbiór faktów, jej uaktywnienie pozwoli na wzbogacenie bazy faktów o nową wiedzę ale z określeniem odpowiednio mniejszego stopnia wiarygodności (tu określanej stopniem niepełności wiedzy IF).

Całościowa analiza problematyki wiedzy niepełnej w systemach wspomagania decyzji połączona z badaniami przeprowadzonymi w ramach tej pracy pozwoliły uformować tezę pracy, zgodnie z którą:

> Metoda analizy skupień wraz z metodą badania stopnia niepełności pozwoli na zwiększenie efektywności wnioskowania w bazach wiedzy z uwzględnieniem niepełności wiedzy.

Głównym celem pracy jest opracowanie metod poprawiających efektywność wnioskowania w systemach wspomagania decyzji dzięki zmianie struktury bazy wiedzy w strukturę skupień reguł podobnych do siebie i uformowaniu dla każdego skupienia jego reprezentanta. Ponadto optymalizacja procesów wnioskowania staje się możliwa dzięki uwzględnieniu niepełności wiedzy. Efektywność wnioskowania mierzona jest zyskiem czasowym wynikającym z braku konieczności przeglądania wszystkich reguł w bazie wiedzy oraz liczbą nowych faktów wygenerowanych w trakcie procesu wnioskowania.

W rozprawie dokonana została analiza efektywności wnioskowania w bazach wiedzy o złożonej strukturze (tj. o dużej liczbie reguł, dowolnej liczbie przesłanek, bez ograniczeń dla typu atrybutów i ich wartości). Uwzględnienie reprezentacji dla wiedzy niepełnej pozwala na zwiększenie efektywności systemów opartych na bazach dziedzinowych. Istniejące podejścia bazujące na teorii zbiorów przybliżonych, teorii Bayesa oraz innych okazują się niewystarczające do tak postawionego problemu. Wykorzystanie zaproponowanych w rozprawie modyfikacji algorytmów analizy skupień oraz metody badania stopnia niepełności wiedzy pozwoli na realizacje procesów wnioskowania efektywnie mimo, ze wiedza początkowa cechowała się niepełnością - co w rzeczywistych systemach wspomagania decyzji jest zjawiskiem dość często spotykanym. Dlatego niniejsza rozprawa ma dostarczyć narzędzia i metody pozwalające na rozwiązywanie problemów tego typu.

1.1 Układ pracy

Rozdział drugi pracy przedstawia ogólną strukturę modułów systemu wspomagania decyzji. Omówiono tu strukturę bazy wiedzy oraz metody jej tworzenia. Zaproponowano także model hierarchicznej struktury bazy wiedzy służący optymalizacji procesów wnioskowania. Trzeci rozdział stanowi poparty przykładami obszerny przegląd istniejących metod i algorytmów wnioskowania w SWD opisanych dotąd w literaturze. W kolejnym rozdziale zaprezentowano istniejące reprezentacje wiedzy niepewnej wraz z formalnymi definicjami oraz przykładem zapisu wiedzy dziedzinowej w każdym z omawianych sposobów. Treść rozdziału piątego składa się z omówienia metod analizy skupień wykorzystywanych w rozprawie wraz z autorskimi modyfikacjami służącymi poprawie efektywności wnioskowania w systemach z wiedzą niepełną. W szczególności uwagę poświęcono omówieniu metody współczynników niepełności wiedzy (IF). Szósty rozdział prezentuje metody oceny efektywności, zarówno algorytmów tworzących skupienia reguł (analiza skupień), jak również efektywności procesu wnioskowania w SE. Przedstawione jest również szacunkowe określenie złożoności obliczeniowej analizowanych algorytmów. Rozdział siódmy to projekt systemu implementującego algorytmy i metody opisane w tej pracy oraz służącego do przeprowadzenia eksperymentów obliczeniowych. Kolejny rozdział przedstawia wyniki eksperymentów uzyskanych w trakcie prowadzonych badań. Eksperymenty te miały na celu wyznaczenie optymalnych wartości parametrów algorytmu grupującego: m.in. wyboru metody tworzenia reprezentanta skupień,

miary odległości pomiędzy skupieniami oraz kryterium łączenia skupień. Przedstawione są również wyniki eksperymentów porównujących omówione w rozprawie metody wnioskowania korzystające z hierarchicznej struktury bazy wiedzy oraz korzystające z reprezentantów skupień. Dokonano również analizy eksperymentalnej przydatności i poprawności zaproponowanej w rozprawie miary do określania stopnia niepełności wiedzy. Pracę kończy podsumowanie porównujące tezę przedstawioną we wstępie z rezultatami uzyskanymi w trakcie przeprowadzonych badań. Na płycie CD dołączonej do pracy umieszczono aplikację jak i przykładowe bazy wiedzy.

Rozdział 2

Systemy Wspomagania Decyzji

W tym rozdziale przedstawione zostaną wybrane zagadnienia z dziedziny systemów wspomagania decyzji (SWD) [158]. Omówiony będzie kompletny opis takiego systemu oraz jego poszczególnych modułów. Zaprezentowana zostanie metoda tworzenia SWD, począwszy od procesu akwizycji wiedzy eksperckiej, a skończywszy na testowaniu, wdrożeniu i ocenie jakości stworzonego systemu.

System ekspertowy (SE) jest definiowany na kilka sposobów:

1. "(...) inteligentny program komputerowy, wykorzystujący procedury wnioskowania do rozwiązywania tych problemów, które są na tyle trudne, że normalnie wymagają znaczącej ekspertyzy specjalistów. Wiedza wraz z procedurami wnioskowania może być uważana za model ekspertyzy, normalnie posiadanej tylko przez najlepszych specjalistów w danej dziedzinie. Wiedza w SE składa się z faktów i heurystyk. Fakty są podstawą bazy wiedzy systemu – informacją, która jest ogólnie dostępna i powszechnie akceptowana przez ekspertów w danej dziedzinie. Heurystyki są zwykle bardziej prywatną informacją" [36].

2. "(...) system komputerowy nawiązujący interaktywny dialog z użytkownikiem za pomocą zgromadzonej wiedzy obejmującej zwykle wąski i specjalistyczny obszar działalności człowieka. Co za tym idzie - system umożliwia na podstawie wiedzy w nim zawartej, zasugerowanie decyzji na podstawie przesłanek, które za każdym razem wprowadza użytkownik. System potrafi również wytłumaczyć na jakiej podstawie podjął taką, a nie inną decyzję (odpowiedź na pytania "jak" oraz "dlaczego") [94]".

3. "(...)program, który przedstawia rozumowanie wraz z wiedzą na temat wąsko zdefiniowanej dziedziny, stworzony z myślą o rozwiązywaniu problemów. System ekspertowy może także spełniać funkcje doradcy człowieka przy podejmowaniu decyzji (udzielanie porad) [73]".

4. "(...) program działający zazwyczaj w wąsko wyspecjalizowanej dziedzinie, spełniający funkcje eksperta w tej dziedzinie. Pozwala on na rozwiązanie wielu problemów, które do tej pory przewyższały możliwości istniejących systemów oraz możliwości obliczeniowe komputerów [97].

Jak widać, każda z tych definicji posiada wspólny mianownik – system ekspertowy jest rozumiany jako program komputerowy, posiadający zebraną wiedzę oraz reguły służące do dowodzenia tez. Autorzy podkreślają wąski zakres wiedzy systemu ekspertowego w celu lepszego zamodelowania fragmentu rzeczywistości. Jednakże od momentu powstania pierwszych systemów ekspertowych moc obliczeniowa komputerów znacznie wzrosła i obecnie udaje się wspomagać decyzje z coraz szerszego zakresu dziedzinowego. Należy również zauważyć, iż raz stworzony system ekspertowy może posłużyć niezliczonej liczbie użytkowników nieposiadających wiedzy eksperckiej. Dzięki temu, synteza wiedzy eksperckiej, może być wykorzystana w znacznie szerszym gronie odbiorców.

SWD stanowią ważną część systemów informatycznych. Przy obecnym szybkim napływie informacji, ich przetwarzanie oraz interpretacja coraz częściej muszą zostać powierzone systemom automatycznym. 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, a masowe przetwarzanie dużej ilości danych jest możliwe w coraz bardziej efektywny sposób.

SWD czasami zwane są również systemami ekspertowymi. Celem ich działania jest wspomaganie lub docelowo – całkowite zastąpienie człowieka-eksperta dziedzinowego w procesie podejmowania decyzji.

Rolą SWD jest realizacja procesu wnioskowania. W ten sposób konkluzje reguł uaktywnionych tworzą nową wiedzę zapisywaną w bazie wiedzy jako nowe fakty. Są one umieszczone w osobnym zbiorze w postaci iloczynu par atrybut-wartość zwanych *deskryptorami*. Dzięki wydzieleniu osobnej części bazy wiedzy, możliwym jest jej wymiana oraz ulepszanie bez ingerencji we wnętrze systemu. Dzięki temu przyspieszony zostaje proces tworzenia nowych systemów wspomagania decyzji.

Aktualnie można również spotkać przedstawicieli SWD stworzonych do celów rozrywkowych i zabawy. Jednym z przykładów może być system Akinator* w którym to zasymulowana jest popularna gra w 20 pytań. Witryna zadając użytkownikowi szereg pytań odgaduje wymyśloną przez niego postać ze świata bajek i kreskówek.

W dalszej części rozprawy pojęcia systemu ekspertowego oraz systemu wspomagania decyzji będą utożsamiane, pomimo różnic jakie sobą prezentują†. Przyjęło się uważać, iż system ekspertowy stanowi syntezę wiedzy eksperta-człowieka zapisaną w sposób elektroniczny. Wiedza ta umożliwia wyciąganie logicznych wniosków poprzez mechanizmy wnioskowania, jednakże dopiero w SWD – wiedza ta stanowi podstawę do wspomagania użytkownika-człowieka w wypracowaniu decyzji w oparciu o znane wcześniej fakty.

Przyjęto, iż system ekspertowy powinien spełniać pięć podstawowych warunków [24]:

1. Poprawność
2. Uniwersalność
3. Złożoność
4. Autoanaliza
5. Zdolność do uaktualniania wiedzy.

System ekspertowy nazywamy poprawnym, jeśli zapewnia on wysoką jakość wyznaczanych decyzji. Jakość ta jest mierzona zwykle w stosunku do człowieka-eksperta, który potwierdza i waliduje wyniki działania systemu. Poprawny system udostępnia wyniki użytkownikowi w rozsądnym czasie oraz dysponuje metodami do naśladowania wiedzy, intuicji, doświadczenia oraz sposobu pracy eksperta.

Uniwersalność SWD to cecha pozwalająca na zdolność rozwiązywania szeregu problemów z danej dziedziny. Przyjęto, iż uniwersalny system nie zawiera sztywnych rozwiązań, lecz pozwala na elastyczne korzystanie ze zbudowanych wcześniej reguł i heurystyk w bazach wiedzy. Aktualnie, pełna uniwersalność jest jeszcze nieosiągalna przez algorytmy komputerowe, jednakże w wyniku badań nad sztuczną inteligencją – jest ona coraz bliższa.

*Dostępny pod adresem http://en.akinator.com/

†M.in. faktu odseparowania bazy wiedzy od mechanizmów wnioskowania, "całościowego" wspomagania decyzji przez SWD oraz modelowania jednego fragmentu rzeczywistości przez SE

Celem powstania systemów ekspertowych było odciążenie człowieka-eksperta od żmudnego procesu analizy i wnioskowania na tych samych danych. Z racji tego, iż ekspert dziedzinowy posiada bardzo obszerną wiedzę, systemy komputerowe naśladujące zachowanie człowieka powinny cechować się odpowiednio dużą bazą wiedzy.

Autoanaliza systemu ekspertowego to możliwość przedstawiania użytkownikowi końcowemu uzasadnień podjętych decyzji. W tym celu system powinien na każdym etapie móc odpowiedzieć na pytania "jak" została wyznaczona aktualnie rozpatrywana konkluzja oraz "dlaczego" użytkownik pytany jest o przesłankę potrzebną do aktywowania jednej z reguł zapisanych w systemie. SWD powinien się również cechować determinizmem oraz możliwością wstecznej analizy ścieżek przeprowadzonego wnioskowania.

Jednym z elementów systemu ekspertowego jest moduł komunikacji z użytkownikiem. Jest on wykorzystywany nie tylko do udostępniania wyników wnioskowania i analizy, lecz również do pozyskiwania nowych faktów i reguł dopisywanych potem odpowiednio do bazy faktów i bazy wiedzy. Ze względu na to, iż użytkownik może wprowadzić wiedzę do systemu, która będzie sprzeczna i niespójna z zapisaną już wiedzą w bazie wiedzy, cała wiedza pozyskiwana od użytkownika musi być weryfikowana. Weryfikacja ta obejmuje zwykle porównanie i analizę nowodostarczonych reguł w stosunku do ich zgodności z aktualnie posiadanymi. Jeśli sprawdzenie wypadnie pomyślnie, reguła może zostać dopisana do bazy wiedzy. W przeciwnym wypadku, należy dokonać rozstrzygnięcia konfliktu spowodowanego przez niezgodność aktualnie posiadanej informacji z nowodopisywaną. SWD powinien rozróżnić sytuację, w której nowe reguły będą po prostu uaktualnieniem starej bazy wiedzy, wtedy należy je po prostu dopisać do bazy wiedzy. Jeśli jednak nowe informacje będą w sprzeczności z elementami dotychczas zapisanymi – należy przeprowadzić analizę konfliktów. W literaturze tematyce analizy konfliktów poświęcono wiele uwagi. Znane są metody zaproponowane przez prof. Zdzisława Pawlaka [118] oraz [30]. Analiza konfliktów nie jest tematem niniejszej rozprawy, zatem nie będzie dalej podejmowana.

2.1 Geneza systemów ekspertowych

Po raz pierwszy system informatyczny zwany systemem ekspertowym powstał na Uniwersytecie Stanforda. Zespół badawczy, który opracował pierwsze systemy ekspertowe składał się m.in. z Edwarda Shortliffe'a, Bruce'a Buchanana, Stanley'a N. Cohena, Randalla Davisa, Williama van Melle'a,

Carli Scott. Uważa się, że były one jednym z pierwszych sukcesów badań nad sztuczną inteligencją [139]. Systemy DENDRAL oraz MYCIN opracowane tamże stały się pierwszymi pełnoprawnymi systemami ekspertowymi. DENDRAL przeznaczony był do identyfikacji struktury molekularnej nieznanych związków chemicznych na podstawie danych otrzymanych w wyniku analiz spektroskopowych. Celem systemu MYCIN była natomiast diagnoza i terapia zakaźnych chorób krwi. W późniejszym czasie, usunięta została z niego wiedza dziedzinowa, a system pod nazwą EMYCIN stał się bazą dla innych zastosowań takich jak SACON (system ekspertowy dla inżynierów konstrukcyjnych), PUFF (analiza i interpretacja badań pulmonologicznych) czy BLUEBOX (wspomagający wybór terapii psychiatrycznej) [140].

Kolejnym etapem w historii SE było opracowanie języka Prolog w roku 1972. Jest to używany do dziś język programowania logicznego umożliwiający stworzenie tzw. systemów szkieletowych, w których mechanizmy wnioskowania oderwane były od bazy wiedzy. Wystarczyło tylko zapisać bazę wiedzy z danej dziedziny w odpowiednim formacie danych i system ekspertowy mógł być użyty od zaraz bez kosztownej i czasochłonnej procedury tworzenia SE od zera. Język Prolog zawiera mechanizmy wnioskowania używające logiki pierwszego rzędu zapisywanych w postaci omówionych wcześniej klauzul Horna[‡].

W latach ’80 ubiegłego wieku nastąpił prawdziwy rozkwit systemów ekspertowych. Wystarczy tu nadmienić, że około dwie trzecie najbogatszych firm uwzględnionych w czasopiśmie Fortune 1000 zastosowało te technologie w swoich codziennych aktywnościach [88].

Najbardziej znanymi narzędziami związanymi z systemami ekspertowymi z tamtych czasów były m.in. Guru (wykonany na podstawie systemu Emycin), Personal Consultant Plus (USA), Nexpert Object, Genesia, VP Expert (USA), Xi oraz Crystal. W ich budowaniu przodowali Amerykanie, Francuzi oraz Brytyjczycy. Wszystkie z nich miały niestety jedną niezaprzeczalną wadę: konstrukcja bazy wiedzy musiała być dokonana przez wprawnego programistę, ze względu na skomplikowany język opisu.

W roku 1986 jednak nastąpił przełom za sprawą systemu Intelligence Service [37] (zwany później ”Pandorą”). Zaimplementowano w nim klasyczny rachunek zdań, a baza wiedzy mogła być tworzona za pomocą języka naturalnego. System zawierał moduł objaśnień i moduł detekcji konfliktów pomiędzy faktami.

Wtedy większość systemów ekspertowych nie korzystała jeszcze z języ-

[‡]Zbiór formuł logicznych w której co najwyżej jeden element jest niezanegowany

ka Prolog. Dopiero od początku lat '90 sytuacja ta ulegała zmianie dzięki popularyzacji i zaadaptowaniu tego języka wśród szerokiej rzeszy badaczy [103, 136].

2.1.1 Obecne systemy ekspertowe

W dzisiejszych czasach systemy ekspertowe stanowią jedną z głównych gałęzi sztucznej inteligencji. Nowe odkrycia oraz wykorzystanie dorobku innych dziedzin wiedzy pozwoliły na stworzenie wielu różnych aplikacji mających zastosowanie w prawie każdej dziedzinie życia.

W artykule [85] autorzy proponują użycie mechanizmów probabilistycznych w celu optymalizacji efektywności wnioskowania w sieciach społecznościowych ze współczynnikami zaufania. Widać tutaj analogię do wnioskowania z użyciem współczynników pewności (omówionych w rozdziale 5.3 na stronie 116), lecz całość używa probabilistycznych mechanizmów wnioskowania w sieciach Bayesa.

W książce [63] znaleźć można intensywne studium przypadku tzw. normatywnych systemów ekspertowych (ang. *normative expert systems*). Autor proponuje stosowanie "analizy decyzyjnej" zamiast tradycyjnych systemów ekspertowych opartych na naśladowaniu zachowania ekspertów. Tradycyjnie, używane są metody heurystyczne lub "ad hoc", które jednakże mogą powielać błędy ekspertów. W odróżnieniu od nich, proponowane przez autora "normatywne SE" korzystają z mechanizmów zachowań normatywnych (czyli trzymania się wypracowanych złotych standardów w każdym przypadku) w odróżnieniu od zachowań deskryptywnych (gdy tych standardów nie zachowujemy). Autor wykorzystuje również nowy sposób zapisu wiedzy, tzw. diagram wpływu (ang. *influence diagram*). Jest to graficzna reprezentacja przekonań, wątpliwości oraz preferencji twórcy SE, która tworzy bazę wiedzy w normatywnym systemie ekspertowym.

Artykuł [52] stanowi obszerne i wyczerpujące studium nad medycznymi SWD (ang. *clinical decision support systems - CDSS*). Znajdują się w nim zarówno opisy systemów historycznych, jak również tych nowoczesnych. Znajduje się tam również wyczerpujący opis metod reprezentacji wiedzy oraz algorytmów wnioskowania użytych w CDSSach (skupiając się na następujących rodzajach: wiedza regułowa, sieci Bayesa, heurystyka, sieci semantyczne, sieci neuronowe, algorytmy genetyczne i oparte na przypadkach).

Pozycja [64] również w sposób wyczerpujący podejmuje problematykę SWD. Znajduje się w niej nowy algorytm o nazwie RETE służący do po-

prawy efektywności wnioskowania w przód. Sposób jego działania sprowadza się do braku konieczności przeglądania całej bazy faktów po uaktywnieniu jednej z reguł (ze względu na drobną zmianę w bazie faktów). Budowana jest dodatkowa struktura (zwana siecią RETE) umożliwiająca szybkie odnalezienie reguł do uaktywnienia.

Pozycja [128] przedstawia zalety wnioskowania opartego na przykładach (ang. *case based*). Autorzy przekonują o wielu zaletach takowego - głównie dzięki możliwości zapisu dodatkowej meta-wiedzy niedostępnej w regułowych systemach wspomagania decyzji. Przedstawiane jest również wnioskowanie w oparciu o tak stworzoną bazę wiedzy.

Korzystając ze stworzonego przez siebie systemu Pathfinder (służącego do diagnozy patologii limfatycznych) autorzy [61] dokonują empirycznego porównania trzech mechanizmów wnioskujących: opartego na twierdzeniu Bayesa oraz zakładającego niezależność hipotez, opartego na teorii Dempstera-Shafera oraz na zmienianiu wartości prawdopodobieństwa szans (ang. *odds–likelihood updating* - w dalszej części artykułu noszącego miano "podejścia związanego z równoległą kombinacją funkcji współczynników pewności CF" ang. *an approach related to the parallel combination function in the certainty-factor (CF) model).* W artykule dowiedziono, że właściwy dobór mechanizmów wnioskujących może (wbrew powszechnie panującej opinii) wpłynąć na skuteczność działania systemu.

Odrębnie traktowane winny być systemy rozmyte oparte na twierdzeniu o zbiorach rozmytych (ang. *fuzzy sets*) [170]. Przegląd aktualnych rozwiązań można znaleźć w artykule [157]. Dzisiejsze systemy ekspertowe tworzone są pod jedno, konkretne rozwiązanie, takie jak kontrola ruchu kolejowego [35], wspomaganie budowy ścieżek w wytwarzaniu półprzewodników [22], modelowanie miast i ich wzrostu [89], ocena ryzyka bankructwa [44], wspomagania planowania militarnego otwarcia ognia [124], diagnostyki medycznej [4] oraz ocena geologicznej stabilności stoku [146], itd.

Istnieją również systemy o bazie wiedzy złożonej z rozmytych reguł "JEŻELI-TO": autorzy [53, 54] zajmują się problemem optymalnego systemu wspomagania decyzji oceniającego ryzyko w siłowni nuklearnej. Ciekawym podejściem jest hierarchiczny model rozmyty [45] służący do optymalnego doboru pracownika. Rozwiązanie to cechuje się synergią algorytmów heurystycznych oraz rozmytego systemu wspomagania decyzji.

Systemy ekspertowe mają szerokie zastosowanie w niemal każdej dziedzinie. Oprócz przedstawionych powyżej, wyróżnić można wiele innych dziedzin wspomaganych przez SE: nadzór sieci telefonicznej na podstawie raportów o uszkodzeniach i zgłoszeń abonenckich (ACE), systemy diagno-

zy medycznej (CASNET), wyznaczanie relacji przyczynowo – skutkowej w diagnostyce medycznej (ABEL), interpretowanie postaci elektrokardiogramów (CAA), identyfikacja struktur cząstek białka (CRYSTALIS), diagnostyka maszyn cyfrowych (DART), interpretacja wyników badań geologicznych przy poszukiwaniu ropy naftowej (DIPMETER ADVISOR), diagnostyka komputerów (FAULTFINDER, IDT), interpretacja wyników pomiarów dla potrzeb chemii (GAL), identyfikacja związków chemicznych metodą emisyjną (GAMMA), wspomaganie badań geologicznych (LITHO), diagnostyka chorób nowotworowych (ONCOCIN), poszukiwanie złóż minerałów (PROSPECTOR), diagnostyka siłowni jądrowych (REACTOR), planowanie eksperymentów genetycznych (MOLGEN, GENESIS, SPEX), kompletowanie sprzętu komputerowego (CONAD, R1, XCON), diagnostyka lokomotyw spalinowych (DELTA), kształcenie lekarzy (Gwidon), szkolenie operatorów siłowni jądrowych (STEAMER), analiza obwodów cyfrowych (CRITTER), analiza układów elektrycznych (EL), analityczne rozwiązanie zadań w zakresie algebry i równań różniczkowych (MAKSYMA), modelowanie układów mechanicznych (SACON) oraz prowadzenie dialogu z maszyną cyfrową w języku naturalnym (INTELLECT) [26].

2.2 Formalna definicja SWD

Formalnie SWD zdefiniowany jest następująco [159]:

$$
\begin{aligned}
SWD &= < U, A, V, f > \\
U &= \{r_1, r_2, \ldots, r_n\} - \text{niepusty, skończony zbiór reguł zapisanych} \\
&\quad \text{w postaci klauzul Horna,} \\
A &= C \cup D = \{a_1, a_2, \ldots, a_m\} \cup \{dec_1, dec_2, \ldots, dec_p\} - \\
&\quad \text{niepusty, skończony zbiór atrybutów warunkowych (C)} \\
&\quad \text{i decyzyjnych (D),} \\
V &= \{V_{(a_1)} \cup V_{(a_2)} \cup \ldots V_{(a_m)} \cup V_{(dec_1)} \cup \ldots \cup V_{(dec_p)}\} - \\
&\quad \text{rodzina zbiorów wartości atrybutów,} \\
f &: U \times A \rightarrow V - \text{funkcja informacji.}
\end{aligned}
$$

Przez r_i oznaczona będzie i-ta reguła w systemie odpowiadająca przyjętej w klasycznych SWD postaci zbioru formuł logicznych, w której co najwyżej jeden element jest niezanegowany. Tak stworzona reguła zwana jest klauzulą Horna [21], gdzie każdy literał z części przesłankowej i decyzyjnej tworzony jest w oparciu o zbiór atrybutów A oraz zbiory wartości

każdego atrybutu $V_a, a \in A$. Para (a, v_a) (gdzie v_a to wartość atrybutu a) budująca przesłanki i konkluzje reguł będzie dalej nazywana deskryptorem $(d_h = (a_j, v_{a_j}))$, dzięki czemu regułę r_i możemy przedstawić następująco: $r_i = d_1 \cdot d_2 \cdot \ldots \cdot d_m \rightarrow dec_1 \cdot dec_2 \cdot \ldots \cdot dec_j$ dla $j \leqslant p$. Deskryptory są zwykle zapisywane w postaci $a_k = v_k$, jednakże spotyka się również równoważny zapis w postaci (a_k, v_{a_k}).

Metoda reprezentacji wiedzy w postaci iloczynu dwójek (a_k, v_{a_k}) nie jest jedynym sposobem na symboliczny zapis wiedzy. Innym podejściem do problemu jest użycie trójki (a_k, rel_k, v_{a_k}), gdzie rel_k oznacza nazwę relacji zachodzącej pomiędzy atrybutem a_k oraz wartością v_{a_k}. Przykładem takiej relacji może być "różne", "mniejsze", "nie większe", itp..

Trzecim ze sposobów zapisu wiedzy jest odniesienie pary atrybut-wartość w stosunku do konkretnego obiektu (reguły). Taki zapis jest symbolicznie oznaczany jako (o_i, a_k, v_{a_k}), gdzie wartość v_{a_k} jest przyporządkowywana atrybutowi a_k dla konkretnego obiektu (reguły) o_i.

Podstawą działania SWD jest baza wiedzy oraz algorytmy wnioskowania [21, 159].

Baza wiedzy jest zwykle dana w postaci zbioru reguł. Każda reguła składa się z części przesłankowej oraz konkluzyjnej połączonych spójnikiem implikacji ($\rightarrow$). Każda z nich stanowi zbiór par atrybut-wartość (deskryptorów) połączonych spójnikiem logicznym "oraz". Zakładając, że mamy dane atrybuty warunkowe `C={Pogoda, Pora roku}` oraz atrybut decyzyjny `D={Iść na basen}` wraz z ich odpowiednimi wartościami

$$\begin{aligned} V_{\texttt{Pogoda}} &= \{\texttt{Ładna, Brzydka}\}, \\ V_{\texttt{Pora roku}} &= \{\texttt{Wiosna, Lato, Jesień, Zima}\}, \\ V_{\texttt{Iść na basen}} &= \{\texttt{Tak, Nie}\} \end{aligned}$$

to przykładowa reguła może wyglądać następująco:

`JEŻELI Pogoda=Ładna ORAZ Pora roku=Lato TO Iść na basen=Tak`

lub też korzystając z zapisu ze spójnikami logicznymi:

`(Pogoda, Ładna)` $\wedge$ `(Pora roku, Lato)` $\Rightarrow$ `(Iść na basen, Tak)`

Korzystając z metody reprezentacji wiedzy w postaci "atrybut-relacja-wartość" kolejną przykładową regułę można by zapisać w następujący sposób:

`JEŻELI Suma opadów <mniejsza niż> 20mm ORAZ Pora roku <równa> Lato TO Czas spędzony na basenie <większy niż> 2h.`

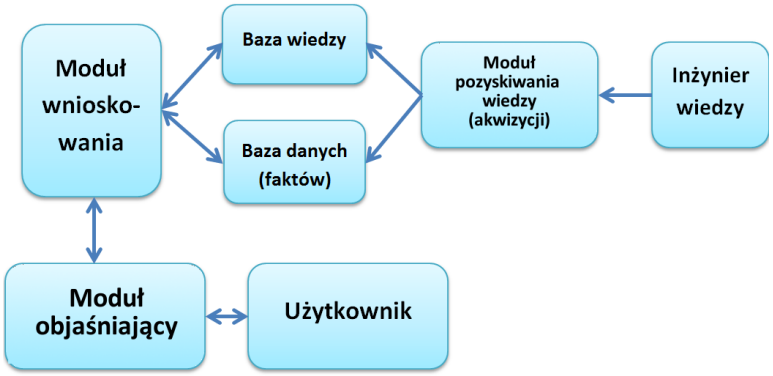
Z kolei dokładając do reguły informacje o konkretnym obiekcie, którego tyczy się wartość, przykładowo, jeśli są dane:

$$\begin{aligned} O &= \{\texttt{Dzisiaj}\}, \\ A &= \{\texttt{Pogoda}\}, \\ V_{\texttt{Pogoda}} &= \{\texttt{Ładna}\}, \end{aligned}$$

informację o tym, że dzisiaj jest ładna pogoda można zapisać następująco: `(Dzisiaj, pogoda, ładna).`

2.3 Architektura systemu ekspertowego

Każdy system ekspertowy składa się z kilku wzajemnie uzupełniających się modułów (rys. 2.1):



Rysunek 2.1: Schemat modułów systemu ekspertowego. Źródło: Opracowanie własne

Baza wiedzy Jest tutaj zawarta cała wiedza zapisana w systemie za pomocą faktów i reguł. Zapis jest uzależniony od wyboru metody reprezentacji wiedzy. Stanowi ona najważniejszą część systemu ekspertowego, od jej jakości oraz pełności zależy skuteczność wnioskowania.

Baza danych (baza faktów) Klasyczna baza danych, która zawiera znane fakty w obrębie dziedziny działania systemu. Niektóre informacje znane *a priori* są dopisywane do bazy faktów przed rozpoczęciem wnioskowania. W trakcie procesu wnioskowania, nowe udowodnione fakty zostają dopisane do tej bazy. Dzięki temu można stosować tzw.

wnioskowanie wielopoziomowe, w którym uprzednio udowodnione fakty stają się przesłankami służącymi do udowodnienia kolejnych faktów.

Moduł wnioskowania Część systemu odpowiedzialna za generowanie konkluzji z wiedzy i faktów. Moduł ten składa się z mechanizmu wyszukującego reguły relewantne, a więc takie, których wszystkie przesłanki są spełnione, a co za tym idzie – w obliczu aktualnego zbioru faktów reguły te mogą zostać uaktywnione w procesie wnioskowania. Szczegółowo proces wnioskowania będzie omówiony w kolejnych rozdziałach.

W pracy [25] odnaleźć można informację, iż moduł wnioskowania składa się z dwóch części. Pierwsza z nich to interpreter reguł, którego zadaniem jest odnalezienie reguł możliwych do uaktywnienia. Druga to maszyna wnioskująca, która używając wybranego algorytmu wnioskowania przeprowadza proces wydobywania wiedzy zgodny z wybranym algorytmem, a więc albo potwierdza postawioną tezę albo po prostu wyprowadza nową wiedzę z systemu.

Moduł pozyskiwania wiedzy Jest to część systemu ekspertowego odpowiedzialna za rozwój i rozbudowę bazy wiedzy. W wyniku różnych metod akwizycji wiedzy eksperckiej, moduł ten dopisuje gromadzoną wiedzę w formacie przyjętym dla danego systemu ekspertowego.

Moduł objaśniający Odpowiedzialny za konwersację z użytkownikiem - zbieranie od niego przesłanek służących do przeprowadzenia procesu wnioskowania. Drugim ważnym zadaniem tego modułu jest umiejętność wyjaśniania w jaki sposób system doszedł do wygenerowanych wniosków oraz dlaczego system pyta użytkownika o wartość danego atrybutu w trakcie wnioskowania. Jest odpowiedzialny zarówno za wprowadzanie danych do systemu, jak i za wyprowadzanie na zewnątrz wniosków systemu. Moduł ten daje użytkownikowi radę, sugestię, lecz nie podejmuje decyzji. Realizuje swoje zadania m.in. poprzez odpowiedź na pytania "jak" oraz "dlaczego" w każdym momencie działania systemu. Odpowiedź na pierwsze z nich pozwala użytkownikowi na prześledzenie ścieżki wnioskowania, która doprowadziła do udowodnienia danej konkluzji ("jak udowodniona została aktualna hipoteza robocza"). Z kolei odpowiedź na pytanie "dlaczego" daje wiedzę w jakim celu system ekspertowy potrzebuje do dalszej

pracy informacji o wartości tego konkretnego atrybutu, o który aktualnie pyta użytkownika. Takie zachowanie jest doskonale widoczne w programie PC-Shell [94].

Ekspertem nazywamy osobę posiadającą rozległą wiedzę i umiejętności dziedzinowe. Przez lata taka osoba gromadziła doświadczenia, dzięki czemu jest nazywana autorytetem i ekspertem dziedzinowym. Do najważniejszych cech eksperta należą wspomniana wcześniej duża i rozległa wiedza, umiejętność jej wykorzystania w praktyce oraz zdolność do uzupełniania wiedzy przez doświadczenie. Zadaniem systemów ekspertowych jest do pewnego stopnia naśladowanie lub też wspomaganie człowieka w procesie podejmowania decyzji.

Etapy tworzenia SWD można zdefiniować następująco:

1. Rozpoznanie problemu, przekazywanie wiedzy dziedzinowej przez eksperta. Studium wykonalności postawionego problemu.
2. Analiza i usystematyzowanie wiedzy. Przygotowanie modelu systemu wraz ze zdefiniowaniem jego granic, danych wejściowych oraz wyjściowych.
3. Zebranie wiedzy od ekspertów.
4. Analiza metod reprezentacji wiedzy i wybór najodpowiedniejszej (m.in. wybór pomiędzy reprezentacjami uwzględniającymi wiedzę niepewną, a zawierającymi tylko wiedzę pewną, rozważenie pozytywów i negatywów każdej reprezentacji w kontekście budowanego systemu, itp.).
5. Implementacja wybranej metody reprezentacji wiedzy. Budowa SWD. Kodowanie systemu.
6. Testowanie systemu i naprawa błędów.
7. Utrzymanie i konserwacja systemu.

Jak widać, przedstawiony model nie różni się bardzo od klasycznego modelu budowy aplikacji. Został on uszczegółowiony w ramach specyficznych potrzeb przy budowie SWD.

2.3.1 Akwizycja danych

Baza wiedzy, będąca kluczowym elementem SWD, musi zostać pozyskana. W początkowej fazie istnienia systemów ekspertowych, wiedza dziedzinowa pozyskiwana była tylko od eksperta dziedzinowego, lecz obecnie wykorzystuje się metody nie wymagające udziału człowieka.

Dzisiejszy dynamiczny rozwój technologii eksploracji danych pozwala na wykorzystanie szeregu automatycznych metod pozyskiwania danych ze zbiorów oraz hurtowni danych. Sieci wielkopowierzchniowych centr handlowych codziennie dostarczają ogromnej ilości danych, co w połączeniu z automatycznymi metodami ekstrakcji wiedzy pozwala im na multiplikowanie zysków i optymalizację zarówno kosztów, jak i sposobów ekspozycji towaru. Znane są metody indukcji reguł decyzyjnych z tablic decyzyjnych (omówione na stronie 2.3.4), metody generowania reguł asocjacyjnych [2], tworzenia drzew klasyfikacyjnych [129] czy wnioskowania z użyciem teorii zbiorów rozmytych [25].

Przed rozwojem automatycznych metod ekstrakcji wiedzy, to inżynier wiedzy przejmował na siebie ciężar syntezy wiedzy eksperckiej. Dominowały wtedy badania nad symbolicznymi technikami uczenia się związanymi z symbolicznymi metodami reprezentacji wiedzy. Dużą popularność zdobyło tzw. *empiryczne uczenie symboliczne*. Polega ono na stworzeniu i zmodyfikowaniu nieznanych wcześniej opisów symbolicznych. Opis ten tworzony jest na podstawie przykładów lub faktów znanych wcześniej. Nie jest wymagana uprzednia wiedza dziedzinowa od osoby, która jest nauczana. Istotny z punktu widzenia procesu uczenia jest za to wybór odpowiednich atrybutów i zbiorów ich wartości, aby użyte przykłady były reprezentatywne i w zadowalający sposób wyeksponowały charakterystyczne cechy danego zagadnienia.

Gdy wiedza pozyskiwana jest bezpośrednio od eksperta, mamy do czynienia z tzw. *uczeniem się na pamięć*. Wiedza jest utrwalana bezpośrednio po jej ustaleniu tak jak to ma miejsce np. przy nauce wiersza na pamięć. Żadne procesy wnioskowania nie są tu zaangażowane. Oczywistym jest, że metoda ta sprawdza się jedynie przy bardzo prostych przypadkach. Rozwinięciem tej metody jest skorzystanie z instrukcji, gdzie współpraca na linii ekspert-inżynier wiedzy powoduje transfer wiedzy dziedzinowej. Ekspert wciela się w rolę nauczyciela i steruje procesem nauczania.

Kolejnym ze sposobów pozyskiwania wiedzy jest *dedukcja*, czyli proces wnioskowania od ogółów do szczegółów. Po uporządkowaniu informacji następuje proces rozumowania. Podejście od drugiej strony, czyli poprzez

indukcję, pozwala na podstawie znanych obserwacji i faktów formułować ogólne hipotezy, które następnie są weryfikowane. Gdy w grę wchodzą znane wcześniej przykłady, dokonywana jest ich generalizacja. Tworzone są ogólne opisy lub też charakterystyki, które są następnie wykorzystywane do opisu całej klasy przypadków. Takie przykłady mogą być określane przez eksperta, pobierane z różnych baz danych, itp.

Rola nauczyciela zostaje znacznie ograniczona w przypadku nauki na podstawie *obserwacji*. Tutaj inżynier wiedzy dzięki swojemu doświadczeniu i intelektowi musi odkryć wiedzę początkowo mu nieznaną. Metoda ta jest stosowana z powodzeniem w technikach grupowania i analizy obrazów. Wyróżnia się tutaj obserwację bierną, gdzie obserwator nie ma wpływu na warunki eksperymentów, może jedynie obserwować oraz obserwację czynną. Przykładem uczenia czynnego na podstawie obserwacji jest metoda iniekcji pakietów w sieciach bezprzewodowych w celu odkrycia hasła zabezpieczeń. Atakujący dokonuje wstrzyknięcia spreparowanych datagramów i na podstawie odpowiedzi uzyskanych od urządzeń dostępowych, jest w stanie odgadnąć hasło szyfrowania.

Ostatnią metodą nauki, jednocześnie najtrudniejszą, jest uczenie się na podstawie *analogii* bazujące na wykorzystaniu doświadczeń wcześniejszych, w poznawaniu nowej dziedziny.

Gdy wiedza przekazywana jest od jednego eksperta, którego wiedza będzie zapisana w systemie ekspertowym, sprawa jest stosunkowo prosta. Jeśli ekspert jest w stanie, to prosimy go o zapisanie swojej wiedzy w formacie reguł postaci *"jeżeli ... to ..."*. Takie podejście jest najwygodniejsze dla projektantów systemu, bo pozwala na wpisanie wiedzy wprost do bazy wiedzy. Bardzo łatwo również stworzyć komentarze w języku naturalnym dla tworzonego systemu. Wadami tego sposobu jest jednak duża trudność dla eksperta w przedstawianiu wiedzy w ten sposób. Bardzo często nie potrafi on podać prawidłowych przesłanek lub też jest ich kilka, często wzajemnie niespójnych. W wyniku takiego przetwarzania wiedzy generowana jest znaczna liczba reguł, które mogą się powtarzać (ekspert nie pamięta, co powiedział).

Drugim podejściem jest próba zapisu reguł z uwzględnieniem współczynników prawdopodobieństwa wpływu konkretnych wartości atrybutów na końcową konkluzję. Niestety, ludzie z natury nie radzą sobie z dokładnym określaniem prawdopodobieństw (jaka jest praktyczna różnica pomiędzy zdaniem "Jutro na 90% będzie padał deszcz" oraz "Jutro na 91% będzie padał deszcz"?). W rzeczywistości pojedyncze cechy mają małe znaczenie przy ustalaniu prawdopodobieństw, dużo ważniejsze są ich zespoły i grupy.

Kolejną metodą pozyskiwania wiedzy od jednego eksperta jest obserwacja jego działania i umieszczania w bazie wiedzy tylko gotowych i sprawdzonych przykładów zastosowania wiedzy w praktyce. Niestety, pomimo faktu, że tak powstała baza wiedzy będzie sprawdzona pod względem merytorycznym, to bardzo mało prawdopodobne jest, że uwzględni wszystkie przypadki działania systemu. Co więcej - zebranie odpowiedniej ilości wiedzy w tym przypadku jest bardzo czasochłonne.

Wiedza pozyskiwana z jednego źródła jest znacznie łatwiejsza do syntezy. Jednakże w zastosowaniach praktycznych zwykle mamy do czynienia z kilkoma ekspertami. Istnieje wiele metod na uzgodnienie ich decyzji tak, aby nie wprowadzać niespójności i sprzeczności w syntezowanej wiedzy np. omówiona poniżej metoda "delficka". Szczegółowe informacje na temat przedstawianego problemu można znaleźć w pracy [163].

Wywiad jako metoda uzyskiwania wiedzy od eksperta nie sprawdza się w przypadku większych grup eksperckich. Oprócz tego, spora część wiedzy szczegółowej i ukrytej zostaje pominięta (ze względu na jej oczywistość w oczach eksperta). Zebranie wszystkich ekspertów razem i przeprowadzenie konferencji – "burzy mózgów" zwykle jednak doprowadza do sytuacji patowej i konfliktowej. Jednakże jeśli już dojdzie do konsensusu, prościej tutaj o uzyskanie wiedzy ukrytej i szczegółowej. Techniki te zostały przedstawione w pracy [144].

W przypadku gdy mamy do czynienia z grupą ekspertów, należy poprosić ich o zajęcie wspólnego stanowiska. Jeśli takie nie zostanie osiągnięte na drodze dyskusji (lub też dyskusja bezpośrednia nie jest możliwa), stosuje się odpowiednio skonstruowane ankiety i kwestionariusze. Każdy ekspert wypełnia taki sam kwestionariusz, a następnie odsyła go w celu porównania z innymi. Po analizie i określeniu wartości średnich, kieruje się informację zwrotną do ekspertów. Odpowiadają oni podając źródła swojej wiedzy, a następnie generowane są zestawienia wiedzy, opinii i źródeł. Każdy z ekspertów ustosunkowuje się do przedstawionego mu raportu i albo zmienia zdanie, albo wskazuje na przyczynę swojego niezmiennego stanowiska. Całą metodę (zwaną "delficką") kończymy w momencie ustabilizowania się wyników. Metoda ta gwarantuje, że opinie ekspertów będą niezależne i, że ostatecznie uda się uzyskać zgodny sąd ekspertów.

Możemy powiedzieć, że metoda delficka charakteryzuje się trzema własnościami:

1. Przeprowadzenie w pełni anonimowej ankiety,

2. Sprawdzanie odpowiedzi na pytania na każdym etapie oraz stała po-

prawa treści pytań (zarówno dopisywanie jak i modyfikacja już istniejących),

3. Sporządzanie raportów z całości przeprowadzonych badań, ale uwzględniając tylko te kwestie, dla których osiągnięto konsensus.

2.3.2 Baza wiedzy

Jakość bazy wiedzy wiernie oddaje jakość całego SWD stanowiąc podstawowy moduł każdego systemu ekspertowego. Wraz z metodami wnioskowania stanowi o faktycznej efektywności systemu. Znajdują się w niej reguły oraz fakty zapisane zgodnie z przyjętą reprezentacją.

Czynnikami określającymi bazę wiedzy są szczegółowość oraz zakres wiedzy. Wysoka szczegółowość oznacza znaczną ilość informacji z danej (najczęściej wąskiej) dziedziny, zakres zaś - rozległość wiedzy.

W dzisiejszych SWD wyróżniamy dwa typy baz wiedzy. Najczęściej stosowane są bazy wiedzy stworzone dla konkretnego problemu, często bardzo specyficznego. Jak można się domyślić bazy takie będą charakteryzowały się wysoką szczegółowością oraz niskim zakresem wiedzy. Pozwala to na skuteczne operowanie w ramach konkretnego problemu poddawanego do rozważania za pomocą SWD. Analogicznie do sytuacji medycznej - będzie to baza wspomagająca lekarza-specjalistę w konkretnej jednostce chorobowej, stosunkowo wąskiej i wybranej spośród wielu innych.

Drugim podejściem jest stworzenie bazy internistycznej. Mamy tu do czynienia z dużym zakresem wiedzy, jednakże o małym stopniu szczegółowości. Sytuację taką można porównać do lekarza pierwszego kontaktu, który powinien posiadać umiejętność diagnozy dużej liczby chorób i przypadków lub też – potrafić określić genezę problemu w celu wysłania pacjenta do specjalisty. Podobnie, w systemie ekspertowym – klasyfikuje się dany problem do bardziej szczegółowego rozważania.

Przy budowie bazy wiedzy należy mieć na uwadze dwie niezbędne cechy. Po pierwsze, baza powinna być wzajemnie niesprzeczna. Oznacza to, że koniunkcja przesłanek (bądź pojedyncza przesłanka w regule) nie powinna generować różnych (sprzecznych) konkluzji. Po drugie, baza powinna być pełna, czyli zawierać wszystkie możliwe przypadki, które przewidywane są przez system.

Niestety, w dzisiejszych zastosowaniach baza wiedzy jest często generowana w sposób automatyczny lub też półautomatyczny z hurtowni bądź też z baz danych. Proces generowania takiej wiedzy jest szybki, jakkolwiek

bardzo często wiedza dostarczona w ten sposób jest wzajemnie sprzeczna, niepełna, a jej użyteczność – niska. Celem tej pracy będzie zaproponowanie sposobu na radzenie sobie z niepełnością wiedzy.

Wyróżnia się dwa podstawowe typy symbolicznej reprezentacji wiedzy:

Reprezentacja deklaratywna, gdzie deklarowane są fakty, stwierdzenia oraz reguły dziedzinowe.

Reprezentacja proceduralna polegająca na zapewnieniu opisu procedur służących do reprezentowania wiedzy o dziedzinie.

Skorzystanie z reprezentacji proceduralnej zwykle zwiększa efektywność działania systemu, przykładowo poprzez zastosowanie właściwych praw matematycznych i fizycznych (np. równań fizycznych). Z drugiej jednak strony, wiedza w postaci deklaratywnej jest łatwiejsza w opisie i formalizacji.

2.3.3 Regułowa reprezentacja wiedzy

Regułowa reprezentacja wiedzy jest najczęstszą i najbardziej intuicyjną formą gromadzenia wiedzy. Dane, w oparciu o które tworzona jest wiedza dla danego SWD bardzo często mają formę tablicy decyzyjnej, w której każdy wiersz składa się ze zbioru pewnych przesłanek (warunków) oraz części decyzyjnej (konkluzji).

System z regułową bazą wiedzy można podzielić na dwa rodzaje ze względu na sposób uzyskiwania ostatecznych konkluzji w procesie wnioskowania:

1. System z regułami prostymi, gdzie konkluzja reguł stanowi wyniki pośrednie. Konieczne jest tutaj wielokrotne analizowanie reguł ze względu na możliwość wystąpienia wnioskowania wielopoziomowego, w którym przesłankami reguł są konkluzje wcześniej wyznaczone.

2. System wykorzystujący reguły złożone, które umożliwiają bezpośrednie uzyskanie ostatecznych konkluzji.

Każdy ze sposobów zapisu reguł ma swoje wady i zalety. Do wad reguł prostych należy głównie konieczność użycia specjalistycznych algorytmów wnioskowania, które potrafią sobie radzić z wnioskowaniem wielopoziomowym. Innymi słowy: konieczne jest uaktywnienie dużej liczby reguł w celu przeprowadzenia poprawnego procesu wnioskowania. Reguły te muszą być również uaktywniane w odpowiedniej kolejności, należy zwrócić uwagę na

tzw. algorytmy doboru reguł – czyli strategie doboru uaktywnianej reguły w sytuacji, gdy wiele z nich może zostać aktywowanych (patrz rozdział 3.2).

Niewątpliwą zaletą systemów z regułami prostymi jest bardzo łatwy proces akwizycji wiedzy oraz aktualizacji bazy wiedzy. Z racji stosunkowo nieskomplikowanej budowy, reguły te mogą zostać zmieniane i wymieniane bez szkody dla całego systemu. Ekspert dziedzinowy ma również ułatwione zadanie w trakcie procesu weryfikacji poprawności reguł.

Systemy wykorzystujące reguły złożone nie wymagają skomplikowanego modułu wnioskowania, ze względu na bezpośredni charakter otrzymywanych konkluzji. Reguły te są dużo bardziej skomplikowane, mają zdecydowanie obszerniejszą część przesłankową, a każda z tych reguł zawiera już ostateczną konkluzję. Wadą takiego podejścia jest trudność formułowania odpowiedniego zbioru reguł oraz złożony sposób jego weryfikacji i uzupełniania.

Do niewątpliwych zalet regułowej reprezentacji wiedzy należy między innymi naturalny sposób zapisu wiedzy. Reguły w postaci "Jeżeli ... to ..." są intuicyjnym sposobem myślenia ludzi, co w łatwy sposób przekłada się na prostotę zapisu. Ponadto, taki sposób zapisu wiedzy nakłada pewne uporządkowanie na bazę wiedzy. Łatwo można wyróżnić część konkluzyjną i warunkową danej reguły, a co za tym idzie – każda reguła jest samodokumentująca się i stanowi niezależną porcję wiedzy możliwą do transferu do innych systemów ekspertowych. W dziedzinach, w których liczba reguł jest duża i niemożliwa do zapamiętania przez człowieka, regułowy zapis wiedzy wymusza pewne uporządkowanie. Dzięki temu możliwe jest odkrycie nowej, nieznanej dotąd wiedzy.

Możliwa jest również modularyzacja bazy wiedzy, a więc rozdział reguł dotyczących wybranego fragmentu opisywanej dziedziny do osobnych źródeł wiedzy (np. osobnych plików). Dzięki temu łatwiej potem inżynierowi wiedzy zarządzać wiedzą, zwłaszcza jeśli reguł jest dużo.

Zapis regułowy wymusza również reprezentację wiedzy w postaci w której formuły logiczne połączone są ze sobą alternatywą. Jest to naturalny i intuicyjny zapis, każda z reguł stanowi o osobnym fragmencie rzeczywistości.

Oddzielna baza wiedzy pozwala na modyfikację i ulepszanie mechanizmów wnioskujących i szkieletu SE bez konieczności przepisywania wiedzy zgodnie z zasadą modułowości. Oprócz tego, możliwym jest tworzenie różnych aplikacji i systemów bazujących na tej samej bazie wiedzy, a posiadających różne zastosowania.

Jak zostanie to przedstawione w późniejszych rozdziałach, do regułowego

zapisu wiedzy można w bardzo prosty sposób wprowadzić informację o niepewności przesłanek, konkluzji oraz całych reguł. Istniejące systemy wykorzystujące współczynniki pewności (ang. *certainty factors*) oraz przedstawiany w tej rozprawie system korzystający ze współczynników niepełności wiedzy są tego przykładem.

Bezsprzecznie, regułowy zapis wiedzy nie pozwala na żadne ustępstwa w stosunku do przyjętych zasad, co jest zaletą w sytuacjach w których człowiek jest podatny na wpływy z zewnątrz (np. w sektorze ubezpieczeniowym lub finansowym).

Ostatecznie, system ekspertowy jest niewrażliwy na kolejność zapisu reguł, podczas gdy człowiek ma tendencję do zwiększania wagi reguł i przypadków najświeższych, dodanych do bazy wiedzy najpóźniej.

Indywidualny zapis reguł, oprócz niezaprzeczalnych zalet, ma również wady. Trudno jest w łatwy sposób dostrzec zależności pomiędzy poszczególnymi regułami występującymi w całym systemie, zwłaszcza gdy tych jest bardzo dużo. Ponadto, trudno jest określić wpływ konkretnej reguły na sposób zachowania się całego systemu jako całości.

Duże systemy regułowe wymuszają konieczność modyfikacji metod wyszukiwania reguł relewantnych w stosunku do aktualnie badanej hipotezy (lub aktualnego zbioru faktów). Proste przeszukiwanie już w systemach o złożoności około 1000 reguł jest nieefektywne. Stąd powstały metody do radzenia sobie z tą niedogodnością [106].

Regułowe systemy ekspertowe nie mają w klasycznej wersji możliwości "odstępstwa od zasad" jak to ma miejsce w przypadku człowieka-eksperta dysponującego tzw. "zdrowym rozsądkiem". Jak powszechnie wiadomo, zdarzają się sytuacje nieprzewidziane przez projektanta, w których to człowiek może postanowić o odstępstwie od zapisanych reguł, podczas gdy system ekspertowy takiej możliwości nie ma. Co więcej, system ekspertowy sam z siebie nie jest w stanie nauczyć się nowych zachowań w trakcie takich sytuacji wyjątkowych.

Regułowy zapis nie sprzyja również sytuacji w której system ekspertowy wie, że problem mu zadany nie leży w jego kompetencjach. W przypadku źle zaprojektowanego modułu komunikacji z użytkownikiem, może minąć wiele czasu zanim system ekspertowy podda się i zwróci informacje o braku możliwości podjęcia decyzji.

2.3.4 Tablica decyzyjna

Regułowy sposób zapisu wiedzy jest stosunkowo wygodny i często spotykany w praktycznych zastosowaniach. Najczęściej dane dziedzinowe są gromadzone w formie tabelarycznej (np w bazach danych). Tablice decyzyjne są odpowiednikiem reprezentacji wiedzy tabelarycznej z uwzględnieniem funkcji wspomagania decyzji.

Z danych tak gromadzonych należy wyindukować wiedzę, która może mieć różne formy zapisu. Dla tablic decyzyjnych można w tym celu skorzystać z algorytmów generowania reguł decyzyjnych, w tym również tzw. reguł minimalnych gwarantujących szybki proces decyzyjny.

Formalnie, system decyzyjny może być również dany w postaci tablicy decyzyjnej. Wtedy system informacyjny DT zdefiniowany jest jako dwójka:

$$DT = (U, A \cup \{d\})$$

gdzie

- U jest niepustym, skończonym zbiorem obiektów,
- A jest niepustym, skończonym zbiorem atrybutów warunkowych,
- $d \notin A$ jest atrybutem decyzyjnym,
- Zbiór V_a jest dziedziną atrybutu $a \in A$.

Definiuje się również funkcję informacyjną

$$f : U \times A \rightarrow V, \forall_{a \in A, x \in U} f(a, x) \in V$$

Jeżeli wiedza prezentowana w tablicy decyzyjnej będzie niespójna, czyli co najmniej dwa elementy ze zbioru uniwersum dostarczają sprzecznych informacji (inaczej mówiąc: opisane są tymi samymi wartościami atrybutów warunkowych, lecz są przydzielone do różnych klas decyzyjnych), wtedy należy skorzystać z jednej z licznych metod usuwania niespójności [119, 148]. Omówienie ich jednakże wykracza poza ramy tej pracy.

Procesu generowania reguł można dokonać w sposób naiwny, przyjmując wiersz w bazie wiedzy jako osobną regułę, lub też skorzystać z algorytmu generowania reguł minimalnych [12].

Algorytm ten przedstawiony zostanie na przykładzie. Mając daną tablicę decyzyjną jak w 2.1 należy najpierw sprawdzić, czy tablica decyzyjna jest spójna[§].

[§]To znaczy, czy dla wierszy z tablicy o tej samej kombinacji wartości atrybutów warunkowych decyzje są sprzeczne

Numer reguły	Pogoda (p)	Ilość pieniędzy (k)	Nastrój (n)	Aktywność (a)
1	Słonecznie	Dużo	Wesoły	Piłka nożna
2	Pochmurno	Mało	Smutny	Kino
3	Pochmurno	Dużo	Wesoły	Kino
4	Słonecznie	Mało	Smutny	Spacer
5	Słonecznie	Mało	Wesoły	Piłka nożna
6	Pochmurno	Mało	Smutny	Spacer

Tabela 2.1: Przykładowa tablica decyzyjna; $\{p, k, n\} \in C;\ a \in D$

Jeśli w tablicy decyzyjnej występuje niespójność, należy wybrać jedną z metod radzenia sobie w takiej sytuacji. W przykładzie skorzystano z metody uogólnionego atrybutu decyzyjnego [148]. Wynik działania obrazuje tabela 2.2.

Numer reguły	Pogoda (p)	Ilość pieniędzy (k)	Nastrój (n)	Aktywność (a)
1	Słonecznie	Dużo	Wesoły	Piłka nożna
2	Pochmurno	Mało	Smutny	Kino, Spacer
3	Pochmurno	Dużo	Wesoły	Kino
4	Słonecznie	Mało	Smutny	Spacer
5	Słonecznie	Mało	Wesoły	Piłka nożna

Tabela 2.2: Spójna tablica decyzyjna

W kolejnym kroku generowana jest macierz nierozróżnialności, która to na przecięciu $i - tej$ kolumny oraz $j - tego$ wiersza zawiera atrybuty warunkowe, które odróżniają te dwie reguły. W tablicy 2.3 znajdują się zakodowane nazwy atrybutów zgodnie z nagłówkiem tabeli 2.2.

U/U	1	2	3	4	5
1	-	p,k,n	p	k,n	k
2	p,k,n	-	k,n	p	p,n
3	p	k,n	-	p,k,n	p,k
4	k,n	p	p,k,n	-	n
5	k	p,n	p,k	n	-

Tabela 2.3: Macierz nierozróżnialności

W tym momencie przystąpić należy do uogólnienia macierzy nierozróżnialności dla każdej klasy decyzyjnej osobno. Jednocześnie usuwane zostają atrybuty rozróżniające obiekty wchodzące w skład tej samej klasy decyzyjnej. Przykładowo dla klasy {Piłka nożna} uogólnioną macierz przedstawia tabela 2.4.

U/U	1	2	3	4	5
1	-	p,k,n	p	k,n	-
5	-	p,k	p,k	n	-

Tabela 2.4: Uogólniona macierz nierozróżnialności dla klasy {Piłka nożna}

Z uogólnionej macierzy nierozróżnialności tworzone są funkcje boole'owskie w taki sposób, iż atrybuty wewnątrz komórki łączone są za pomocą alternatywy, natomiast komórki ze sobą – za pomocą koniunkcji [120]. Funkcja jest minimalizowana stosując podstawowe prawa algebry Boole'a [141]:

$$\begin{aligned} f_{MG}(A, \{\text{Piłka nożna}\}, 1) &= (p \vee k \vee n) \wedge p \wedge (k \vee n) \\ &= (pp \vee pk \vee pn) \wedge (k \vee n) \\ &= ppk \vee pkk \vee pkn \vee ppn \vee pkn \vee pnn \\ &= pk \vee pkn \vee pn \\ &= pk(1 \vee n) \vee pn \\ &= pk \vee pn \end{aligned}$$

$$\begin{aligned} f_{MG}(A, \{\text{Piłka nożna}\}, 5) &= (p \vee k) \wedge (p \vee k) \wedge n \\ &= ppn \vee pkn \vee pkn \vee kn \\ &= pn \vee pkn \vee kn \\ &= pn(1 \vee k) \vee kn \\ &= pn \vee kn \end{aligned}$$

Mając daną formułę minimalną funkcji boolowskich sięgamy do oryginalnej tabeli decyzyjnej i wypisujemy reguły:

$$f_{MG}(A, \{\text{Piłka nożna}\}, 1) = pk \vee pn$$

- IF p=Słonecznie AND k=Dużo THEN a=Piłka nożna
- IF p=Słonecznie AND n=Wesoły THEN a =Piłka nożna

$$f_{MG}(A, \{\text{Piłka nożna}\}, 5) = pn \vee kn$$

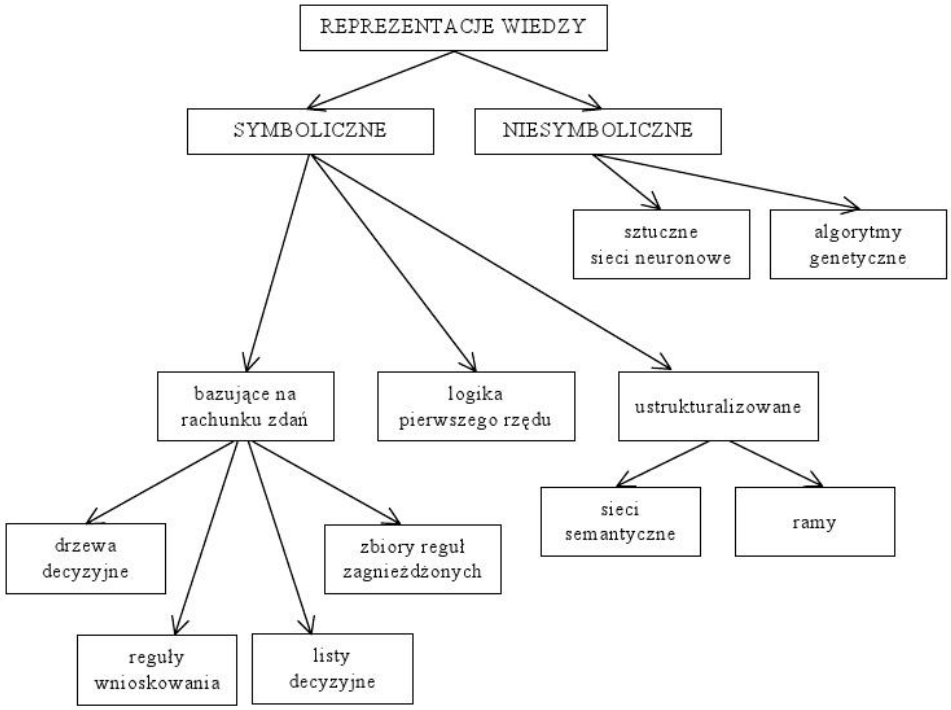
- IF p=Słonecznie AND n=Wesoły THEN a =Piłka nożna
- IF k=Mało AND n=Wesoły THEN a=Piłka nożna

Cały proces powtarzamy dla wszystkich klas decyzyjnych. Dzięki temu otrzymujemy zestaw reguł minimalnych. Aktualnie, algorytm ten [143] jest traktowany jako zbyt wolny w przypadku bardzo dużych zbiorów danych. Rozpatrywane są podejścia przybliżone i aproksymacyjne [14, 50, 79, 101, 115, 119, 125], opierające się na sieciach neuronowych [74] oraz inne jednakże omówienie ich wykracza poza zakres tej pracy.

2.3.5 Inne metody reprezentacji wiedzy

Najbardziej intuicyjna reprezentacja regułowa nie jest jedyną możliwością organizacji wiedzy w bazach wiedzy. W przeciągu wielu lat doskonalenia systemów ekspertowych poszerzyły się możliwości w zakresie reprezentacji wiedzy (patrz rys. 2.2, źródło: [67]).

Regułowy zapis wiedzy jest bardzo wygodnym sposobem organizacji wiedzy, jednakże często niewystarczającym. Dynamiczny rozwój programowania obiektowego oraz obiektowych baz danych wymusza również inne podejście do tego problemu. Pewnym sposobem naśladowania "myślenia obiektowego" jest używanie sieci semantycznych [90]. Sieci te opierają się na wzajemnych powiązaniach i oznaczaniu relacji pomiędzy węzłami. Formalnie, sieć jest zdefiniowana jako skończony, etykietowany, acykliczny graf skierowany, w którym węzły reprezentują obiekty fizyczne (np. pies, człowiek) lub koncepcje (czynności, wydarzenia, np. spotkanie, posiadanie). Łuki łączą ze sobą obiekty oraz ich deskryptory reprezentując różnego rodzaju relacje. Sieć semantyczna jest również definiowana jako uporządkowana trójka: $S =< P, T, R >$, gdzie P oznacza zbiór pojęć (wierzchołków grafu) stanowionych przez nazwy obiektów O, nazwy cech C oraz nazwy wartości cech V, T zbiór typów relacji (zbiór gałęzi grafu), zaś R to zbiór relacji $R \subset P \times T \times P$. Wyróżnia się kilka typów relacji: $O \times O$ to relacja określona na zbiorze obiektów będąca typu "część-całość" ("IS A") lub też relacja podrzędności "IS PART OF"; relacja typu $O \times C$, czyli przysługiwanie obiektom pewnych wartości cech; $V \times C$ - relacja postaci "jest wartością cechy"; $V \times V$ będąca relacją uporządkowania oraz $O \times V$ - relacja typu "posiada wartość cechy".



Rysunek 2.2: Metody reprezentacji wiedzy

Sieci semantyczne wraz z ontologiami stanowią w tej chwili ciekawy sposób reprezentacji wiedzy w sieciach społecznościowych. Szczegółowo o tych zagadnieniach traktuje praca [84].

Kolejnym sposobem reprezentacji wiedzy, znanym z logiki, są rachunki perceptów oraz predykatów [25]. Wnioskowanie w systemie predykatowym oparte jest na aksjomatach rachunku zdań i metodzie dowodów założeniowych. W języku tym występują kwantyfikatory matematyczne. Język predykatów został zaimplementowany w języku programowania Prolog. Powstał on w celu automatycznej analizy języków naturalnych, jest jednak językiem ogólnego zastosowania, szczególnie dobrze sprawdzającym się w programach związanych ze sztuczną inteligencją. Prolog w przeciwieństwie do większości popularnych języków programowania jest językiem deklaratywnym.

Wiedzę reprezentować można również w postaci niesymbolicznej. Metody te bazują na obserwacji natury i korzystają z mechanizmów znanych z przyrody. Jednym z przedstawicieli takiego zapisu są sztuczne sieci neuronowe. Symulują one zachowanie żywych komórek nerwowych, ich połączenia oraz propagację sygnałów elektrycznych. Sieci neuronowe w sposób dynamiczny dostosowują się do zmieniających się warunków poprzez zmianę

współczynników wagowych połączeń nerwowych. Wiedza reprezentowana jest jako sposób przekształcania impulsów wejściowych, poprzez warstwę pośrednią neuronów o eksperymentalnie dobieranym schemacie połączeń, aż do warstwy wyjściowej. Propagacja sygnałów odbywa się poprzez przypisanie wagi do każdego połączenia, a następnie wyliczenie wartości wektora wyjściowego. Sztuczne sieci neuronowe bardzo łatwo dają się "uczyć", dzięki temu bardzo dobrze przystosowują się do nowych, nieznanych wcześniej przypadków [80].

Istnieje również wiele modeli hybrydowych, jak choćby opisany w pracy [23]. Integruje on samoorganizujące się mapy (SOM), sieci neuronowe, algorytmy genetyczne i reguły rozmyte w celu przewidywania wyników sprzedaży obwodów drukowanych. Można w nim odnaleźć grupowanie reguł rozmytych i optymalizację parametrów rozmywania. Wspomniany już wcześniej pakiet Sphinx również zawiera narzędzie o nazwie Neuronix, które tworzy wiedzę dla SWD w postaci sieci neuronowej.

Inną technikę reprezentacji wiedzy stanowią tzw. algorytmy genetyczne, które umożliwiają przekazywanie następnym generacjom wiedzy o całym gatunku. Wiedza ta jest zapisana w genach.

Obszerny przegląd metod reprezentacji wiedzy oraz aktualnych trendów w konstruowaniu systemów ekspertowych można znaleźć w pracy [142].

2.4 Hierarchiczna modyfikacja bazy wiedzy

Efektywność wnioskowania w SWD uzależniona jest przede wszystkim od szybkości znajdowania reguły relewantnej oraz możliwości aktywowania reguł mając do dyspozycji odpowiednią bazę faktów. Niestety, w klasycznych SWD proces odnajdywania reguł jest stosunkowo długi ze względu na konieczność liniowego przeglądania całej bazy wiedzy w poszukiwaniu reguł relewantnych.

Aby skrócić czas wnioskowania potrzebne są techniki pozwalające jak najszybciej przeszukiwać zbiór reguł – kandydatów do uaktywnienia. Zastosowanie metod grupowania reguł podobnych do siebie pozwala na podział dużego zbioru reguł na mniejsze grupy reguł. Utworzenie reprezentantów dla grup pozwoli potem na ich szybkie przeszukiwanie. Zastosowanie metod hierarchicznych (omówionych w dalszej części pracy) w ramach metod grupowania, pozwoli dodatkowo zoptymalizować cały proces wyszukiwania reguł dzięki możliwości użycia efektywnych technik przeszukiwania struktur drzewiastych, w szczególności binarnych. Autor w tym zakresie dokonał

analizy efektywności zarówno hiearchicznych jak i niehierarchicznych metod grupowania w odniesieniu do dokumentów tekstowych i ich szybkiego wyszukiwania, czego efektem są prace [69–72, 107–112, 160, 162].

W literaturze spotyka się już rozwiązania opierające się na budowie struktury hierarchicznej w grupach [106, 161]. Jednakże w proponowanych dotychczas podejściach nie był brany pod uwagę problem niepewności i niepełności wiedzy, co uwzględnione będzie w przypadku proponowanego rozwiązania.

W rozprawie tej proponuje się zmodyfikowanie klasycznego SWD i poddanie go procesowi grupowania w celu zwiększenia efektywności wnioskowania. Zysk z takiego zabiegu będzie dwojaki: szybkość znajdowania i uaktywniania reguł będzie poprawiona dzięki grupowaniu reguł, jednocześnie grupowanie reguł umożliwi wyznaczenie skupień reguł o największym stopniu pokrycia zbioru faktów.

Aby taka modyfikacja mogła mieć miejsce, należy uzupełnić definicję SWD o dodatkowe struktury danych. Tak uzupełniona definicja przedstawia się następująco:

$$mSWD = < U, A, V, f, F_{sim}, Tree >$$

$F_{sim} : U \times U \Rightarrow R|_{([0...1])}$ - funkcja podobieństwa.

$Tree = \{w_1, \ldots, w_{(2n-1)}\} = \cup_{(i=1)}^{(2n-1)} w_i$

-drzewo reguł i grup reguł budowane przez algorytm grupujący.

$w_i = \{D_i, f_{sim}, k_i, l_i\}$, gdzie $D_i = \{d_1, \ldots, d_m\}$

- zbiór deskryptorów wchodzących w skład i-tej grupy,

k_i, l_i – węzły tworzące i-tą grupę.

Wprowadza się hierarchiczną strukturę reguł dzięki której odnajdywanie reguły relewantnej będzie znacznie szybsze. Autor zmodyfikował strukturę hierarchiczną bazy wiedzy znaną z literatury [106] w celu wprowadzenia dodatkowych informacji służących do poprawnego zamodelowania niepełności wiedzy.

Wprowadzona została również koncepcja współczynnika niepełności wiedzy, która pozwoli na skuteczne modelowanie niepewności i niepełności wiedzy. Koncepcja ta wraz z opisem, przykładami zapisu oraz modyfikacją mechanizmów wnioskowania zostanie omówiona szczegółowo w rozdziale 5.3.

2.5 Podsumowanie

W dobie powszechnej komputeryzacji i minimalizacji kosztów procesów przemysłowych, technologicznych, itp. SWD stają się niezbędnym narzędziem pozwalającym na realizację założonych celów. W rozdziale omówiono zarówno podstawowe definicje, obowiązujące zapisy formalne, zaproponowane w literaturze metody reprezentacji wiedzy (z naciskiem na reprezentację regułową), architekturę takich systemów oraz metody ich tworzenia.

Rozdział 3

Wnioskowanie w systemach wspomagania decyzji

Aby system ekspertowy spełniał należycie swoją rolę, reguły, fakty oraz wiedza zapisana w nim powinny być możliwie najpełniejsze. System ekspertowy na podstawie znanych wcześniej faktów i reguł jest w stanie wyznaczać kolejne fakty dopisywane potem do bazy danych.

Taki mechanizm zwany *wnioskowaniem* korzysta z podstawowych praw logicznych, zwykle logiki dwuwartościowej [48]. Zgodnie z polskim logikiem K. Ajdukiewiczem:

"Wnioskowanie jest procesem myślowym, w którym na podstawie mniej lub bardziej stanowczego uznania przesłanek dochodzimy do uznania wniosku, którego dotychczas nie uznawaliśmy wcale bądź uznawaliśmy mniej stanowczo; przy czym stopień stanowczości uznania wniosku nie przewyższa stopnia uznania przesłanek."

Systemy ekspertowe wykorzystują regułę *modus ponens*, inaczej zwaną "regułą odrywania":

$$\frac{(A \Rightarrow B), A}{B}$$

Reguła ta mówi o tym, że jeżeli z przesłanki A wynika logicznie wniosek B oraz przesłanka A jest prawdziwa, to wtedy prawdziwy jest również wniosek B.

Wyróżnia się trzy rodzaje wnioskowania: wnioskowanie w przód (ang. *forward inference, forward chaining, data driven, event driven*), wnioskowa-

nie wstecz (ang. *backward inference, backward chaining, goal driven, expectation driven*) oraz wnioskowanie mieszane [97, 102].

W celu lepszego zrozumienia dalszych rozważań, zakłada się iż dokładna baza reguł to taka baza, w której reguły tworzone są w oparciu o klasyczną logikę dwuwartościową wraz z dwoma stałymi logicznymi: prawda oraz nieprawda. Wprowadza się również pojęcie tzw. *elementarnej bazy reguł*, czyli bazy wiedzy w której nie występują równocześnie warunki niedopytywalne w postaci prostej p oraz zanegowanej $\neg p$. Wnioskowanie na takich bazach będziemy nazywać wnioskowaniem elementarnym dokładnym [102].

Ponadto, dla większości systemów ekspertowych zakłada się prawdziwość tzw. "reguły zamkniętego świata". Zgodnie z nią, zakłada się, że prawdą jest tylko to, co wynika z reguł bazy reguł oraz faktów znajdujących się w bazie faktów. Założenie to funkcjonuje również w języku programowania Prolog stosowanym w modelowaniu systemów ekspertowych [104].

Autor w tej rozprawie nie ogranicza proponowanego systemu do modelu zamkniętego świata. Dzięki temu dopuszczane jest dopisywanie nowych faktów i odpytywanie użytkownika o fakty, które potrzebne są do przeprowadzenia procesu wnioskowania.

3.1 Rodzaje wnioskowania

Wnioskowanie jest procesem prowadzącym do udowodnienia (bądź nie) konkluzji, jeśli wiemy, że przesłanki są prawdziwe. Jak napisano wcześniej, wyróżnia się trzy główne typy wnioskowania: wnioskowanie w przód, wnioskowanie wstecz i wnioskowanie mieszane.

3.1.1 Wnioskowanie w przód

Jest inaczej nazywane wnioskowaniem sterowanym faktami lub wnioskowaniem progresywnym [66]. Algorytm wnioskowania jest stosunkowo prosty. Korzystając z dostępnej bazy wiedzy i bazy faktów należy przeglądać dostępne reguły i sprawdzać, czy wszystkie przesłanki konkretnej reguły są spełnione. Jeśli tak, reguła taka zostaje uaktywniona i jej konkluzja dopisana zostaje do bazy faktów. Warunkiem stopu dla algorytmu wnioskowania w przód jest sytuacja w której nie możliwym jest już uaktywnienie żadnej reguły lub też gdy w bazie faktów znajdzie się hipoteza, którą należało dowieść.

Jak można łatwo zauważyć, wnioskowanie tego typu może doprowadzić do lawinowego przyrostu liczby nowych faktów w bazie danych. Po-

nadto, w przypadku dużych systemów ekspertowych proces uaktywniania dużej liczby reguł i szybki przyrost faktów może doprowadzić do drastycznego obniżenia efektywności systemu. Z drugiej jednak strony, wnioskowanie w przód może przyczynić się do pożądanego zwiększenia początkowo skąpej bazy faktów.

Algorytm wnioskowania w przód można przedstawić w następujący sposób:

Algorytm 1: Wnioskowanie w przód

Dane: B:=Baza wiedzy; F:=Baza faktów; h:=hipoteza do udowodnienia
Rezultat: Hipoteza udowodniona?
while *h - nieudowodnione oraz są jeszcze reguły do uaktywnienia* **do**
  Wyznacz podzbiór reguł R z bazy wiedzy B możliwych do uaktywnienia;
  Wybierz regułę ze zbioru R i ją uaktywnij;
  F:= F + konkluzja wybranej reguły;
end
if *h udowodnione?* **then**
  return *True*;
else
  return *False*;
end

3.1.2 Wnioskowanie w tył

Zwane jest inaczej wnioskowaniem sterowanym celem lub wnioskowaniem regresywnym, ponieważ do działania wymaga podania hipotezy h, którą należy dowieść. Ten typ wnioskowania daje zwykle znacznie szybsze rezultaty, ponieważ nie ma tutaj potrzeby uaktywniania wszystkich możliwych reguł, a jedynie tych, które są absolutnie niezbędne do udowodnienia zakładanej hipotezy.

Algorytm rozpoczyna swoje działanie od sprawdzenia, czy cel wnioskowania znajduje się w bazie faktów. Jeśli tak, cel zostaje udowodniony a algorytm kończy swoje działanie. W przeciwnym wypadku, przeszukujemy dostępne reguły w poszukiwaniu takiej, której konkluzją jest poszukiwany cel. Wybieramy tę regułę zgodnie z przyjętą strategią i jej przesłanki stają się podcelami w procesie dowodzenia. Przeszukujemy bazę faktów w celu sprawdzenia, czy podcele nie znajdują się w niej. Jeśli algorytm ich nie znajduje, po raz kolejny poszukujemy reguł, których konkluzjami są podce-

le. Proces powtarzamy aż wyczerpią się możliwe do uaktywnienia reguły lub wszystkie podcele zostaną udowodnione, a tym samym – pierwotna hipoteza zostaje udowodniona.

W zastosowaniach praktycznych wnioskowanie wstecz jest znacznie szybsze, ponieważ na etapach pośrednich nie udowadnia tak wielu hipotez, jak wnioskowanie w przód. Wnioskowanie w tył niestety nie nadaje się jednak do udowadniania kilku hipotez jednocześnie.

Algorytm wnioskowania w tył przedstawia się następująco:

Algorytm 2: Wnioskowanie w tył

Dane: B:=Baza wiedzy; F:=Baza faktów; h:=hipoteza do udowodnienia
Rezultat: Hipoteza udowodniona?
while *h - nieudowodnione oraz są jeszcze reguły możliwe do zastosowania* **do**
  Wyznacz podzbiór reguł R z bazy wiedzy B możliwych do uaktywnienia, których konkluzje stanowią aktualnie rozpatrywany cel;
  Wybierz regułę ze zbioru R i ją uaktywnij;
  F:= F + konkluzja wybranej reguły;
  if *h udowodnione?* **then**
    return *True*;
  else
    h:=przesłanki reguły koniecznej do udowodnienia aktualnego celu;
    Rozpocznij wnioskowanie wstecz dla nowego podcelu;
  end
end
if *h udowodnione?* **then**
  return *True*;
else
  return *False*;
end

3.1.3 Wnioskowanie mieszane

Zarówno wnioskowanie w przód, jak i wnioskowanie regresywne mają swoje wady i zalety. W trakcie prac nad systemami ekspertowymi postanowiono połączyć obie te metody w celu ich optymalizacji [66]. Wnioskowanie mieszane opiera się na stosowaniu dodatkowej meta wiedzy w postaci meta

reguł (nazywanych potocznie regułami dla reguł), które to pozwalają modułowi wnioskowania na przełączanie się pomiędzy wnioskowaniem w przód i wstecz. Dzięki temu mechanizmowi zakłada się, iż hipoteza udowadniana zostanie dowiedziona efektywniej. Istotną cechą wnioskowania mieszanego jest fakt, iż przy przełączaniu się pomiędzy rodzajami wnioskowania za hipotezę główną przyjmuje się ciągle hipotezę postawioną przez użytkownika na początku działania systemu.

System, w którym zastosowano wnioskowanie mieszane przed rozpoczęciem samego procesu dokonuje wczytania metawiedzy informującej maszynę wnioskującą o preferencjach wyboru metody dowodzenia. Praktycznie wyróżnia się po prostu dwa oddzielne podmoduły wnioskowania, każdy z nich realizujący dowodzenie hipotez w zaimplementowany wcześniej sposób. Dzięki metaregułom następuje przełączenie sposobu wnioskowania w trakcie samego procesu.

W implementacjach systemu ekspertowego przyjmuje się np. podział bazy reguł na dwie części – jedną służącą do przeprowadzenia wnioskowania progresywnego, a drugą – regresywnego. Określa się również priorytet każdego z tych rodzajów wnioskowania.

Wydawać by się mogło, że optymalnym sposobem będzie wyprowadzenie całej możliwej wiedzy z części wnioskowania progresywnego, a następnie użycie jej do udowodnienia hipotezy podanej przez użytkownika (wnioskowanie wstecz). Zwykle jest to prawdą, jednakże istnieją specyficzne bazy wiedzy w których ta sytuacja nie ma miejsca [97].

Wnioskowanie mieszane jest bardzo rzadko stosowane w systemach ekspertowych ze względu na konieczność dostarczenia dodatkowej metawiedzy o regułach. Oprócz tego, jednoczesna implementacja dwóch rodzajów wnioskowania wraz z mechanizmem przełączającym znacząco komplikuje działanie systemu.

3.2 Strategie doboru reguł

W trakcie przeprowadzenia wnioskowania może się zdarzyć sytuacja, w której więcej niż jedna z reguł może zostać uaktywniona. W takim przypadku mechanizm wnioskowania musi zadecydować, którą z reguł ostatecznie aktywować.

Znane są następujące metody sterowania wnioskowaniem (zwane również strategiami "doboru reguł" lub "metodami rozwiązywania konfliktów") [19, 97, 127]:

Strategia świeżości (ang. *recency*) polega na wybraniu reguły, która dopisana została najpóźniej do bazy wiedzy. Stosuje się ją dla baz o dużej dynamice, w których nowodopisywana wiedza jest często wykorzystywana i szybko traci na aktualności. Dopiero, gdy wśród najświeższych reguł nie uda się odnaleźć odpowiedniej, rozpatrywane są reguły starsze.

Strategia blokowania (ang. *refractoriness*) jest domyślnie stosowana w zdecydowanej większości systemów ekspertowych wraz z jedną z pozostałych metod doboru reguł. Każda reguła, która została już wykorzystana w danym przebiegu wnioskowania, zostaje zablokowana do ponownego użycia. Dzięki temu wyeliminowana jest sytuacja, w której system ustawicznie posługuje się jedną z wybranych reguł uaktywniając ją, co nie wnosi żadnej nowej wiedzy do systemu.

Strategia specyficzności (ang. *specificity*) uaktywnia najpierw te reguły, które zawierają większą liczbę przesłanek. Preferowane są bardziej specyficzne, specjalizowane reguły o większej liczbie potwierdzonych przesłanek. Dzięki temu teoretycznie, za każdym razem będzie wybierana reguła o lepszym dopasowaniu do aktualnej sytuacji. W przypadku, gdy kilka reguł ma taką samą liczbę przesłanek, wybiera się reguły o mniejszej liczbie zmiennych.

Strategia pierwsza reguła na liście (ang. *textual order*) jest prostą strategią, w której uaktywniana jest pierwsza reguła na liście zgodnie z kolejnością ich zapisu w bazie wiedzy.

Strategia przypadkowości (ang. *random order*) — uaktywniona zostaje reguła losowo wybrana spośród reguł możliwych do uaktywnienia. Zwykle strategia ta używana jest w ostateczności.

3.3 Algorytmy wnioskowania

Omówione powyżej rodzaje wnioskowania są ogólnymi algorytmami implementowanymi w większości systemów. Niestety, w przypadku, gdy system ekspertowy złożony jest z więcej niż 100 reguł zdarza się, iż ponad 90% czasu działania sprowadza się do wyszukiwania reguł możliwych do uaktywnienia mając dany zbiór [38].

Proces wnioskowania można rozbić na trzy podprocesy:

Dopasowywanie W kroku tym sprawdzana jest część warunkowa każdej reguły z bazy wiedzy pod kątem dopasowania do aktualnego zbioru faktów. Wszystkie reguły możliwe do uaktywnienia tworzą tzw. zbiór reguł możliwych do uaktywnienia (ang. *conflict set*).

Wybór Spośród wszystkich reguł możliwych do uaktywnienia, wybierany jest podzbiór faktycznie uaktywnianych reguł. Sposób wyboru jest różny w zależności od konkretnej implementacji systemu ekspertowego, jednakże najbardziej popularne metody zostały omówione we wcześniejszym rozdziale tej pracy.

Uaktywnienie, czyli inaczej dopisanie części konkluzyjnej uaktywnianych reguł do zbioru faktów.

Bardzo często sprowadza się problem odnajdywania reguły relewantnej w stosunku do aktualnie rozpatrywanego zbioru faktów do problemu dopasowania wzorca (ang. *pattern matching*) [165]. Ma to miejsce ze względu na ten sam cel tych dwóch zadań. Przeanalizujmy regułę oraz przykładowy zbiór faktów:

```
Reg. 1:
JEŻELI (pogoda,słoneczna)
ORAZ (temperatura,wysoka)
ORAZ (samochód,zatankowany)
WTEDY (Jedź nad wodę,Tak)

Zbiór faktów: {{pogoda,słoneczna), {samochód,zatankowany)}
```

Widać tutaj wyraźnie, że dowolny algorytm dopasowywania wzorca powinien odnaleźć fragment reguły, który dokładnie pokrywa się ze zbiorem faktów.

Już w początkowych latach działania systemów ekspertowych problem skutecznego i szybkiego dopasowywania został dostrzeżony. Pierwszą znaczącą poprawą wydajnościową klasycznych algorytmów wnioskowania był zaproponowany przez Forgy'ego w 1979 algorytm RETE [39]. Jego szczegółowe omówienie znajduje się w kolejnym podrozdziale.

Algorytm ten buduje sieć węzłów, z których każdy (prócz korzenia) odpowiada wzorcowi (lub też zbiorowi deskryptorów) występującemu w części przesłankowej reguły z bazy wiedzy. Każda ścieżka od korzenia do liścia odpowiada regule z systemu ekspertowego. Każdy węzeł posiada zapisaną listę faktów, które odpowiadają wzorcowi w nim zapisanym. Wnioskowanie

w takiej strukturze polega na odwiedzaniu węzłów i oznaczaniu ich jako potwierdzone (zgodne ze zbiorem faktów). Po potwierdzeniu całej ścieżki (dotarciu do korzenia), reguła zostaje uaktywniona, a nowe fakty – dopisane do systemu. W kolejnym etapie system sprawdza, czy nowodopisane fakty nie spowodują możliwości uaktywnienia dodatkowych reguł.

Algorytm TREAT [96] jest rozwinięciem algorytmu RETE w którym autorzy starali się zmniejszyć zajętość pamięciową pierwowzoru. W tym celu niektóre z węzłów (pośrednich) nie są pamiętane. Podobnie algorytm GATOR [59] stara się optymalizować algorytm RETE poprzez modyfikację liczby ścieżek wchodzących do węzłów pośrednich sieci.

Kolejnym przykładem optymalizacji algorytmów wnioskowania jest algorytm LEAPS [11]. Przyspieszenie działania tegoż algorytmu polega na implementacji dodatkowej struktury danych wskaźników na reguły możliwe do uaktywnienia w konkretnym przebiegu algorytmu. Po dopisaniu nowych faktów, algorytm decyduje, czy rozpocząć przeszukiwanie reguł od nowa, czy dokończyć poprzedni proces wyszukiwania reguły zdolnej do uaktywnienia. Szczegółowe omówienie algorytmu znajduje się w dalszej części pracy. Daniel P. Miranker w swoim opracowaniu * stara się udowodnić wyższość ostatniego algorytmu. Przytacza informacje o wysokiej złożoności obliczeniowej algorytmów RETE i TREAT, która to została poprawiona asymptotycznie w algorytmie LEAPS. Ponadto, złożoność pamięciowa została również zmniejszona do wartości liniowej. Badania te zostały powtórzone i zweryfikowane w pracy [164].

3.3.1 Algorytm Rete

Algorytm RETE [39] należy do grupy algorytmów wnioskujących w przód. Został on zaprojektowany aby zwiększyć szybkość wnioskowania kosztem złożoności pamięciowej. W większości przypadków, wzrost prędkości wobec naiwnej implementacji wnioskowania progresywnego jest kilkukrotny. Teoretycznie jego wydajność nie zależy od liczby reguł w systemie. Jak się jednak okazało w praktyce, w dużych systemach ekspertowych problemem jest ogromne zapotrzebowanie na pamięć operacyjną [10]. Późniejsze modyfikacje i całkiem nowe podejścia zostały zaprojektowane właśnie w celu wyeliminowania tej niedogodności.

W klasycznej implementacji algorytmu wnioskowania wprzód, moduł wnioskowania dopasowuje części warunkowe każdej reguły do aktualnie rozpatrywanego zbioru faktów aby ocenić, czy reguła nadaje się do uaktyw-

*http://www.cs.utexas.edu/~miranker/treator.htm

nienia. Jeśli reguła zostaje uaktywniona, jej konkluzja zostaje dopisana do zbioru faktów i cały proces wnioskowania rozpoczyna się od początku (ponieważ być może nowe reguły będą mogły być uaktywnione po dostarczeniu nowej wiedzy). Twórca algorytmu RETE zauważył, że uaktywnienie jednej reguły powoduje tylko drobną zmianę w zbiorze faktów, co nie powinno powodować rozpoczęcia procesu wnioskowania od początku, ale tylko uwzględnienie tej drobnej zmiany w tym procesie.

Zasada działania algorytmu RETE sprowadza się więc do śledzenia zmian w zbiorze faktów i odpowiedniej drobnej modyfikacji listy reguł możliwych do uaktywnienia w każdej iteracji wnioskowania.

Algorytm RETE buduje sieć powiązań będącą w istocie drzewem (czyli acyklicznym grafem skierowanym z wyróżnionym węzłem korzenia). W grafie tym występują różne rodzaje wierzchołków (węzłów):

Węzeł Rete to inaczej korzeń, punkt startowy grafu.

Węzeł typu (ang. *Type Node*) jest węzłem zawierającym informacje o nazwie atrybutu (faktu). Dzięki tym węzłom możliwe jest bezpośrednie przejście do wartości danego atrybutu bez przeszukiwania wszystkich deskryptorów systemu. Jak można się domyślić, liczba tych węzłów będzie równa liczbie atrybutów warunkowych występujących w całym systemie ekspertowym.

Węzeł α którego celem jest przyporządkowanie konkretnej wartości danego atrybutu. Węzły tego typu będą węzłami atomowymi, tj. takimi w których zapisana jest dokładnie jeden deskryptor. Dzięki temu mechanizm dopasowywania do wzorca będzie działał znacznie sprawniej.

Węzeł β będący złączeniem dwóch węzłów typu α, a więc - połączeniem kilku różnych deskryptorów z części warunkowej.

Węzeł terminalny (liść) zawierający informacje o numerze porządkowym reguły. W momencie dojścia mechanizmu wnioskującego do liścia, odpowiednia reguła zostaje uaktywniona, a jej konkluzja dopisana do zbioru faktów.

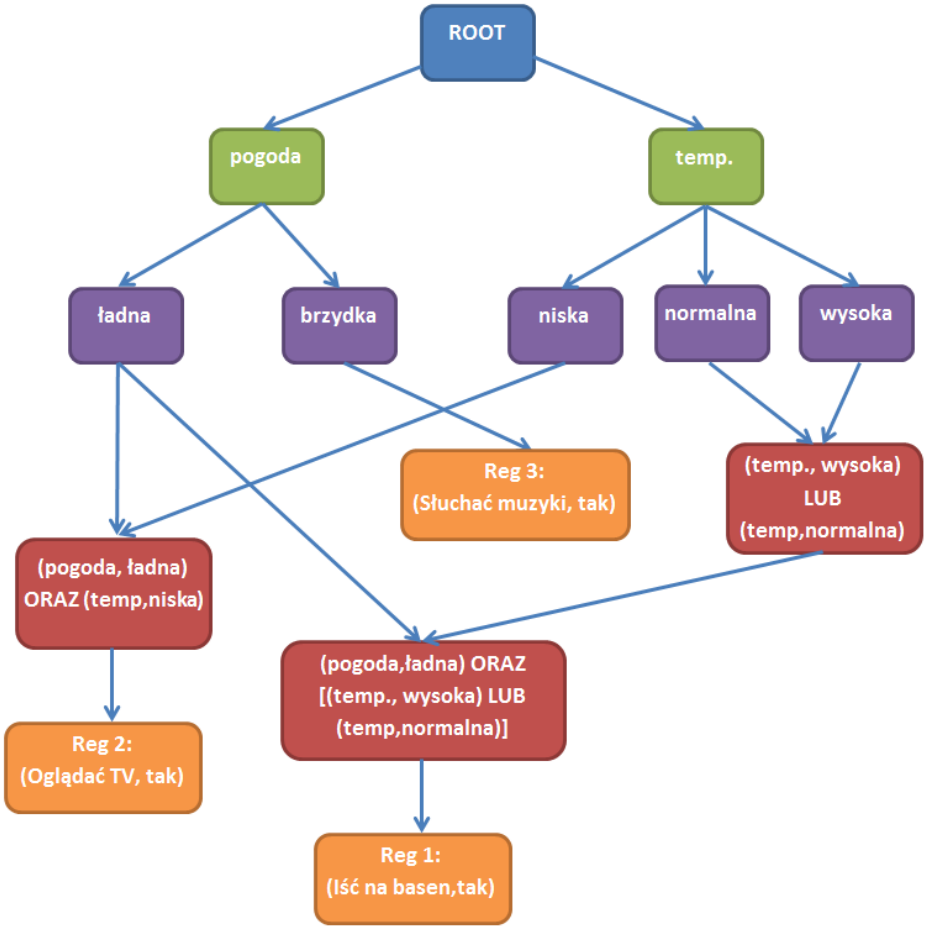
Przykładowo, dla trzech reguł:

1. Jeżeli (pogoda,ładna) ORAZ (temperatura, wysoka) LUB (temperatura,normalna) WTEDY (Iść na basen,tak)

2. Jeżeli (pogoda, ładna) ORAZ (temperatura,niska) WTEDY (Oglądać TV,tak)

3. Jeżeli (pogoda,brzydka) WTEDY (Słuchać muzyki,tak)

graf RETE został przedstawiony na rysunku 3.1. Kolor niebieski oznacza węzeł początkowy, zielony – węzły typu, fioletowy – węzły α, czerwony – wezły β oraz pomarańczowy – węzły terminalne.



Rysunek 3.1: Graf algorytmu RETE

Relacje możliwe do zapisania w węzłach typu α to początkowo tylko relacje równości (np. (pogoda,ładna)). W późniejszych modyfikacjach zbiór relacji uległ rozszerzeniu.

Wnioskowanie w takiej sieci jest stosunkowo proste. Rozpoczyna się ono w korzeniu grafu. Następnie algorytm przechodzi do odpowiednich węzłów

typu. Proces ten możliwy jest do zrównoleglenia, jednakże tylko na etapie węzłów typu. Późniejsze rozgałęzienia i wzajemne połączenia pomiędzy węzłami powodują, iż algorytm RETE bardzo trudno poddaje się paralelizacji [96].

3.3.2 Algorytm Treat

Algorytm RETE był znacznym osiągnięciem w przyspieszaniu wnioskowania w SWD. Niestety, pomimo niewątpliwych zalet, posiadał on również wady, które ujawniały się w implementacjach. Jedną z nich jest, wspomniana już wcześniej, tzw. eksplozja kombinatoryczna zapotrzebowania na pamięć operacyjną w celu poprawnego zapisania wszystkich węzłów sieci. Drugą poważną wadą algorytmu RETE jest mała możliwość zrównoleglania obliczeń [151].

W celu wyeliminowania powyższych wad autorzy [96] zaproponowali nową wersję algorytmu. Główną różnicą w stosunku do poprzednika jest brak konieczności pamiętania węzłów typu β. Są one zbędne dla tego algorytmu, ponieważ wszystkie powiązania są przechowywane w postaci wektorów w węzłach α i łączone ze sobą w razie potrzeby na bieżąco. Drugą zmianą implementacyjną jest użycie wektora wartości atrybutów w węzłach typu α zamiast oddzielnych węzłów tego typu. Dzięki temu znacznie zmniejsza się zajętość pamięciowa algorytmu.

Podobnie jak algorytm RETE, TREAT jest przystosowany nie tylko do dodawania faktów w trakcie wnioskowania, ale również do ich dynamicznego usuwania w trakcie tego procesu. Niekonieczny jest kosztowny proces przeliczania dopasowywania wszystkich węzłów typu α do nowopozyskanej wiedzy.

Szczegółowy sposób działania algorytmu omówiony jest w pracy [96].

3.3.3 Algorytm Leaps

Algorytm LEAPS [11] korzysta z pamięci stosowej w celu optymalizacji szybkości wnioskowania. Wszystkie fakty zostają odłożone na stosie w kolejności ich napływania. Następnie, zostają ściągane jeden po drugim i dopasowywane do poszczególnych reguł. Dopasowanie jest realizowane za pomocą szybkiego kojarzenia po nazwie atrybutu i typie oczekiwanej wartości atrybutu. Niestety, odbywa się to iteracyjnie zgodnie z listą wszystkich atrybutów danego systemu. Optymalizacje zaproponowane przez autora sugerują użycie jednokierunkowych funkcji skrótu (ang. *hash function*) w celu przy-

spieszenia tego procesu. Po odnalezieniu atrybutu i jego wartości, proces jest albo przerywany (niezgodność ze wzorcem) i brana jest kolejna reguła z listy, albo też – kontynuowany od miejsca znalezienia dopasowania. Dzięki temu nie przeglądana jest cała lista reguł po raz kolejny.

Ważną zaletą jest fakt, iż LEAPS umożliwia uaktywnienie reguły przed tym jak wszystkie znane fakty zostaną dopasowane do reguł w systemie. Algorytm LEAPS w momencie znalezienia pierwszej dopasowanej reguły uaktywnia ją. Dzięki temu uzyskano przyspieszenie działania w rzeczywistych systemach.

Kolejnym elementem odróżniającym prezentowane podejście od poprzednich jest znaczna optymalizacja pamięciowa. Nie ma tu konieczności przechowywania węzłów pośrednich typu β. Co więcej – algorytm ten nie przechowuje informacji o wszystkich faktach koniecznych do udowodnienia części przesłankowej reguły, lecz korzysta z tzw. "leniwej ewaluacji" (ang. *lazy evaluation*) faktów polegającej na zapisywaniu ich dopiero wtedy, gdy jest to konieczne.

Jak wspominano wcześniej, każdy zmieniany (dopisywany lub usuwany) element ze zbioru faktów jest przechowywany na stosie. Dodatkową informacją dopisywaną do każdego z nich jest znacznik czasu. Jak można łatwo zauważyć, kolejność elementów odkładanych na stosie jest zgodna z kolejnością ich modyfikacji w trakcie algorytmu wnioskowania.

Po odłożeniu na stos danego faktu następuje cykl wykonywania reguł. Wybierany jest zawsze fakt ze szczytu stosu. Nazywany jest on "obiektem dominującym" (ang. *dominant object (DO)*). Dzięki niemu filtrowana jest baza wiedzy w celu wyboru reguł zawierających dany fakt. Baza ta jest przed wykonywaniem algorytmu sortowana zgodnie ze strategią specyficzności (a więc wg malejącej liczby przesłanek). Po przejrzeniu całej bazy reguł, jeśli którakolwiek z nich jest możliwa do uaktywnienia, jej konkluzja dopisywana jest na szczyt stosu faktów i proces powtarzany jest od początku. Jeśli system nie odnajdzie żadnej reguły, która jest w pełni pokryta – brany jest kolejny fakt ze stosu i proces powtarza się aż do jego wyczerpania.

3.4 Poprawność bazy wiedzy

Baza wiedzy większości systemów ekspertowych nie jest budowana przez jednego człowieka. Ponadto, gromadzenie wiedzy (jak zostało to pokazane w rozdziale 2.3.1) jest procesem, który trwa stosunkowo długi czas. W je-

go trakcie możliwe jest popełnienie błędów, które skutkować będą brakiem poprawności w bazie danych. Większość z mechanizmów systemów ekspertowych działa poprawnie tylko dla niesprzecznych baz wiedzy, co czyni proces testowania (walidacji) bazy wiedzy niezmiernie istotnym.

W procesie testowania wykrywa się reguły zbędne, sprzeczne, pochłaniające, reguły z niepotrzebnym warunkiem oraz reguły zapętlone [97].

3.4.1 Sprzeczności w regułach

Sprzeczność w regułowych bazach wiedzy będzie rozpatrywana w kilku kategoriach [102]. Po pierwsze, mamy do czynienia z regułami zewnętrznie samosprzecznymi:

`JEŻELI Pogoda=Ładna ORAZ Pora roku=Lato TO Pogoda=Ładna`

Reguła będzie nazywana samosprzeczną, gdy przynajmniej jeden deskryptor występuje w niej zarówno w części przesłankowej, jak i konkluzyjnej. Jest to najprostszy typ sprzeczności wykrywany dość często w sposób automatyczny.

Drugim typem sprzeczności jest sytuacja w której istnieją dwie reguły, w których w części warunkowej pierwszej z nich występuje deskryptor d_i występujący również w części konkluzyjnej drugiej reguły i vice versa:

```
1:JEŻELI Pogoda=Ładna ORAZ Pora roku=Lato TO Iść na spacer=Tak
2:JEŻELI Pogoda=Ładna ORAZ Iść na spacer=Tak TO Pora roku=Lato
```

Reguły takie nazywane są zewnętrznie bezpośrednio sprzeczne. Wariantem tej sytuacji jest pośrednia sprzeczność reguł, w czasie gdy istnieje podstawienie reguły m do innej reguły, tej zaś do innej, itp. ostatecznie prowadzące do powyższej sprzeczności.

Specyficznym przypadkiem są reguły, których część decyzyjna jest sprzeczna:

```
1:JEŻELI Pogoda=Ładna ORAZ Pora roku=Lato TO Iść na spacer=Tak
2:JEŻELI Pogoda=Ładna ORAZ Pora roku=Lato TO Iść na spacer=Nie
```

Sprzeczność wystąpi tutaj tylko w przypadku, gdy zbiór wartości atrybutu `Iść na spacer` jest ograniczony do wartości {`Tak, Nie`}.

3.4.2 Nadmiarowość w regułach

W specyficznych przypadkach może się zdarzyć, że jedna z reguł pochłania drugą:

```
1:JEŻELI Pogoda=Ładna ORAZ Pora roku=Lato TO Iść na spacer=Tak
2:JEŻELI Pogoda=Ładna TO Iść na spacer=Tak
```

Widać wyraźnie, że reguła nr 1 jest zbędna, ponieważ zawsze w przypadku spełnienia części warunkowej reguły nr 1, spełniona jest również część warunkowa reguły nr 2, natomiast w drugą stronę taka sytuacja nie zachodzi. Część konkluzyjna obu reguł jest zgodna.

Szczególnym przypadkiem nadmiarowości w regułach są tzw. niepotrzebne warunki:

```
1:JEŻELI Pogoda=Ładna ORAZ Pora roku=Lato TO Iść na spacer=Tak
2:JEŻELI Pogoda=Śnieg ORAZ Pora roku=Lato TO Iść na spacer=Tak
```

Jeśli zbiór wartości atrybutu `Pogoda` składa się wyłącznie z {`Ładna, Śnieg`}, wtedy atrybut ten jest całkowicie zbędny. Sytuacja ta jest skutecznie wykrywana i usuwana przez indukowanie reguł minimalnych omówione w poprzednim rozdziale.

3.5 Podsumowanie

Żaden z przedstawionych algorytmów nie jest przystosowany do radzenia sobie z problemem wiedzy niepełnej. Algorytmy RETE oraz TREAT w sytuacji, gdy zbiór faktów nie umożliwia uaktywnienia żadnej reguły nie pozwolą na zakończenie procesu wnioskowania sukcesem. Narzut czasowy w budowaniu odpowiednich sieci powiązań będzie w tym przypadku całkowicie nieuzasadniony. Co więcej, zbudowana sieć jest trudna do modyfikacji w taki sposób, aby podać użytkownikowi końcowemu informacje jakich faktów brakuje do możliwości uaktywnienia przynajmniej jednej reguły z systemu. Algorytm LEAPS wydaje się być w tym momencie lepszym, jednakże jego największa zaleta – leniwa ewaluacja – sprawia, że reguły, którym do uaktywnienia brakuje małej liczby przesłanek, zostaną zepchnięte w głąb pamięci i zapomniane. Algorytm ten również nie poda żadnych informacji użytkownikowi odnośnie nowych faktów koniecznych do wprowadzenia do systemu w celu przeprowadzenia procesu wnioskowania.

Proponowane przez autora tej rozprawy podejście również sprowadza się do modyfikacji płaskiej, regułowej bazy wiedzy. Autor proponuje użycie algorytmów analizy skupień w celu grupowania reguł w bazie wiedzy. Stworzona hierarchiczna struktura, z pozoru jest podobna do tych tworzonych przez przedstawione algorytmy. Pomoże jednak w szybkim odnajdywaniu

skupień reguł o największym stopniu pokrycia w stosunku do aktualnego zbioru faktów. Proponowane rozwiązanie będzie szczegółowo przedstawione w rozdziałach 5.3 oraz 7.

Rozdział 4

Reprezentacja wiedzy niepewnej

W rzeczywistych zastosowaniach bardzo rzadko zdarza się tak komfortowa sytuacja, gdy wiedza zawarta w bazie wiedzy systemu ekspertowego jest niesprzeczna, spójna i pewna*. Łatwo można wyobrazić sobie sytuację, w której ekspert-człowiek nie jest do końca przekonany o słuszności jakiegoś twierdzenia, ale nie wie dlaczego uważa je za niepewne. Co więcej, reguły, którymi kieruje się ekspert (np. w medycynie czy ekonomii) są rozmyte, przybliżone, nieprecyzyjne i trudne do zastosowania w formie opisanej we wcześniejszych rozdziałach.

Wiedzą niepewną będziemy określać wiedzę przekazaną przez eksperta, która może nie zawsze pokrywa się z pełną wiedzą na dany temat. W większości przypadków ta wiedza sprawdza się w rzeczywistości, jednak czasem bywa niepełna i niepewna. W przypadku pozyskiwania wiedzy od wielu ekspertów nie jest powiedziane, że wszyscy mają jednakowy pogląd na dany temat. Mało tego, specyfika problemu analizowanego przez eksperta może być na tyle trudna do opisania, że jedyne co ekspert może zrobić to określić stopień swojego subiektywnego przekonania o spełnialności tej wiedzy w rzeczywistości. Z niepewnością w wiedzy wiążą się także tzw. pojęcia nieostre oraz po prostu wiedza niespójna.

Eksperci, jak i zwykli ludzie zwykle wolą posługiwać się pojęciami nieostrymi, takimi jak "często", "zwykle", "czasami", "rzadko". Pojęcia te w bardzo trudny sposób modeluje się przy użyciu klasycznych technik budowy bazy wiedzy. W 1944 r. Ray Simpson [99] przeprowadził eksperyment, w którym poprosił 355 studentów szkół wyższych i liceów o wypełnie-

*Czyli na pewno prawdziwa.

Nazwa	R. Simpson (1944)	M. Hakel (1968)
Always (Zawsze)	99	100
Very often (Bardzo często)	88	87
Usually (Najczęściej)	85	79
Often (Często)	78	74
Generally (Zwykle)	78	74
Frequently (Wielokrotnie)	73	72
Rather often (Raczej często)	65	72
About as often as not (Tak samo często jak nie)	50	50
Now and then (Od czasu do czasu)	20	34
Sometimes (Czasami)	20	29
Occasionally (Okazjonalnie)	20	28
Once in a while (Raz na jakiś czas)	15	22
Not often (Nieczęsto)	13	16
Usually not (Zwykle nie)	10	16
Seldom (Niepowszednio)	10	9
Hardly ever (Prawie nigdy)	7	8
Very seldom (Bardzo rzadko)	6	7
Rarely (Rzadko)	5	5
Almost never (Prawie nigdy)	3	2
Never (Nigdy)	0	0

Tabela 4.1: Badanie wartości współczynników przekonania

nie krótkiej ankiety. Ankieta składała się z takich właśnie nieprecyzyjnych pojęć, a zadaniem ankietowanych było przypisanie im wartości "przekonania" będących współczynnikami od 0 do 100. W 1968 r. Milton Hakel powtórzył ten eksperyment, a ich wyniki przedstawia tabela 4.1.

Jak widać, systemy ekspertowe muszą sobie radzić w jakiś sposób z problemem niepewności wiedzy. Niepewność ta może pochodzić nie tylko od błędnych wskazań eksperta, ale również np. z powodu użycia nieprecyzyjnej aparatury pomiarowej czy braku możliwości wykonania jakiegoś pomiaru.

4.1 Podstawowe pojęcia

Pojęcie nieostre będzie rozumiane w dalszej części pracy jako pojęcie, którego nie da się przyporządkować bezpośrednio do klasy "prawda/fałsz". Przykładowe pojęcia nieostre przedstawione są w tabeli 4.1. Do tego typu pojęć zaliczamy również stwierdzenia np.: "wystarczające ciśnienie opon"

czy "odpowiednia temperatura ciała". Aby wnioskowanie było możliwe, każde z takich pojęć należy przekształcić za pomocą metod radzenia sobie z niepewnością wiedzy do postaci odpowiedniej dla przyjętej metody reprezentacji wiedzy.

Pojęcia niespójne to takie, które wprowadzają sprzeczność w bazie wiedzy. Z niespójnością bazy wiedzy mamy do czynienia w sytuacji, gdy dla takiego samego zbioru atrybutów warunkowych system proponuje dwie różne wartości decyzji. Szczegółowo zagadnienie to omówione jest w rozdziale 3.4 na stronie 46.

Niekompletność (niepełność) wiedzy to sytuacja, w której nie otrzymujemy pełnej i wyczerpującej informacji o rozpatrywanym problemie. Istnieje wiele metod, zarówno statystycznych jak i innych pozwalających na uzupełnienie braków w danych (np. poprzez użycie średniej wartości, mediany, czy mody dla wartości kategorycznych) [5, 33, 131, 132]. Niepełność wiedzy w kontekście wnioskowania w SWD rozumiana jest jako sytuacja, w której nie istnieje dostatecznie duża liczba par (a_i, v_{a_i}) (deskryptorów) wchodzących w skład zbioru faktów aby aktywować jakąkolwiek z reguł wchodzących w skład bazy wiedzy.

Niepewność wiedzy to termin używany w dalszej części rozprawy dla oznaczenia konkluzji dopisanych do zbioru faktów w wyniku uaktywnienia reguł, których nie wszystkie przesłanki są spełnione zgodnie z założeniami autorskiego systemu omówionego w części badawczej niniejszej pracy.

W literaturze spotyka się kilka typów metod radzenia sobie z niepewnością wiedzy. Są to najczęściej: sieci Bayesa, współczynniki pewności CF, teoria Dempstera-Shafera oraz zbiory i wnioskowanie rozmyte i przybliżone. Każda z nich będzie omówiona w tym rozdziale.

4.2 Wiedza dziedzinowa

W zależności od sposobu przyjęcia reprezentacji wiedzy, do dyspozycji są różne metody radzenia sobie z wiedzą niepełną. Na podstawie przykładowej wiedzy dziedzinowej przedstawianej poniżej, omówione zostaną wybrane metody reprezentacji wiedzy niepełnej wraz ze sposobami na radzenie sobie z niepełnością wiedzy.

Jan Kolaż startuje w maratonie rowerowym. Jak każdy rozsądny zawodnik wie, że sporo rzeczy może pójść nie tak: począwszy od awarii roweru, zbyt forsownego treningu przed maratonem, po wypadek na trasie. Awaria roweru zdarza się stosunkowo rzadko, ale jednak Janek nie kończy jednego

maratonu na dziesięć z tego powodu. Ostatnimi czasy trasy maratonu są coraz lepiej przygotowane, dlatego wypadki na trasie to około 5% wszystkich startów.

Wynik w maratonie zależy bezpośrednio od tych czynników. Janek może zająć pierwsze miejsce, może być w ścisłej czołówce lub też – poza podium. Jeśli wszystko sprzysięgnie się przeciwko naszemu zawodnikowi, wtedy rzadko jest w czołówce, o wygranej nawet nie myśląc. Awaria roweru i forsowny trening to również zabójcze połączenie, ale z racji doświadczenia – Jankowi udaje się wtedy czasami wygrać zawody, a przynajmniej wielokrotnie stanąć na podium. Prawidłowy trening podwyższa te szanse o 10 procent.

Inaczej ma się sprawa w przypadku, gdy rower będzie świetnie przygotowany i nie ulegnie awarii. Jeśli dodamy do tego umiarkowany trening, to wygrana jest prawie pewna i Janek osiąga pierwsze miejsce w 8 przypadkach na 10, raz przegrywa z kretesem. Wypadek na trasie jednak powoduje, że statystyki nie są już tak imponujące: 6 wygranych i po równo miejsc w czołówce i poza podium na 10 startów.

Najgorsza sytuacja ma miejsce, gdy rower jest sprawny, ale człowiek nie. W tym przypadku psychika nie wytrzymuje i gdy na trasie zdarzy się wypadek, Janek na pewno przegra. Jeśli nie, to tak samo często zajmuje pierwsze miejsce, jak nie oraz ma 20% szans na bycia na podium.

Od wyniku maratonu zależy premia Janka. Jeśli sponsor ma dobry humor, to zarówno pierwsze miejsce jak i ścisła czołówka przyniosą dużą premię. Pierwsze miejsce Janka i zły dzień sponsora, to już tylko 70% szans. Pudło i zły dzień – jeszcze gorzej, bo tylko w połowie przypadków premia zostanie przyznana. Miejsce poza podium z racji przepisów wewnętrznych grupy sponsorskiej nigdy nie przyniesie premii. Sponsor jest jednak przychylny naszemu kolarzowi i w 3 przypadkach na 5 ma dobry humor.

Jaka jest szansa na to, że Janek otrzyma premię w zawodach?

Przedstawiona wiedza dziedzinowa została opracowana na podstawie wywiadu z ekspertem. Jak widać, znajdują się w niej zarówno określenia precyzyjne, ostre ("20% szans", "raz na dziesięć startów", itp.) jak również nieprecyzyjne, nieostre: "czasami", "wielokrotnie", itp.

W powyższym tekście również mamy komfortową sytuację ze względu na określenie "szans" dla prawie każdej możliwej kombinacji wydarzeń. Dzięki temu możliwe jest zbudowanie modeli w oparciu o tak przedstawioną bazę wiedzy. Nie wiemy jedynie, jaka jest szansa Janka na prawidłowy trening.

Niepewność ta została wprowadzona celowo, aby pokazać w jaki sposób poszczególne podejścia radzą sobie z problemem wiedzy niepełnej.

4.3 Wnioskowanie probabilistyczne, sieci Bayesa

Pierwszym sposobem zapisu wiedzy niepewnej jest skorzystanie z rachunku prawdopodobieństwa, a docelowo z teorii Bayesa do zbudowania sieci bayesowskich.

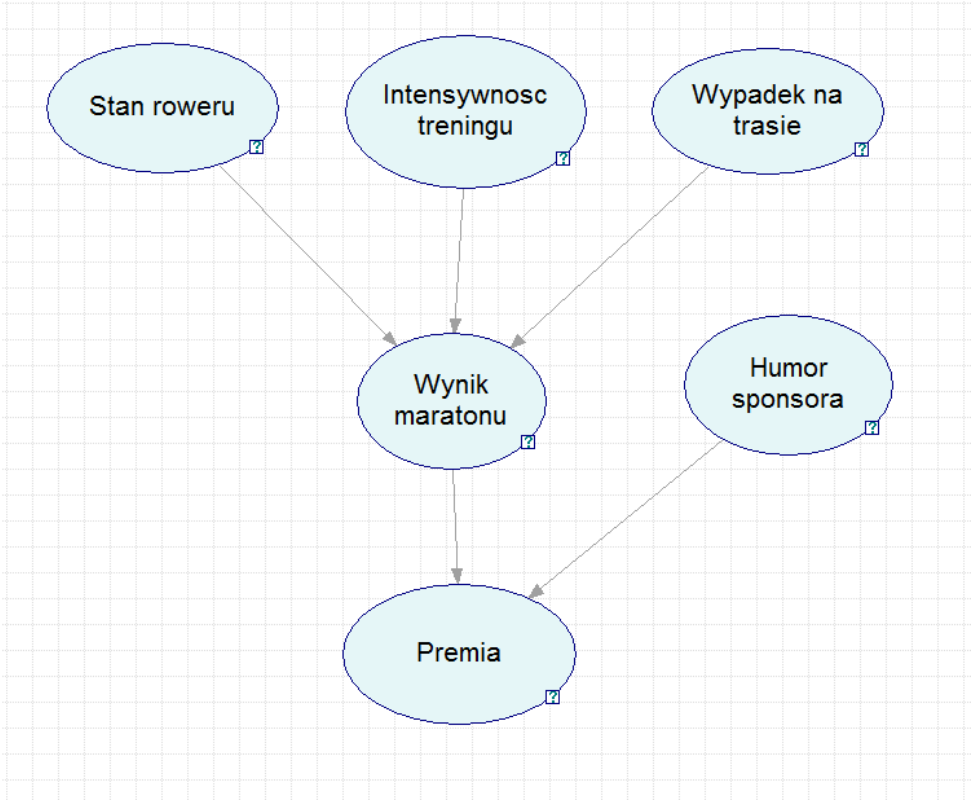
Sieć Bayesa [15] jest przedstawiana jako skończony, acykliczny graf skierowany, zbudowany na podstawie prawdopodobieństw warunkowych. Jest również określana probabilistycznym modelem graficznym, siecią przekonań lub siecią przyczynowo-skutkową reprezentującą zbiór zmiennych wraz z ich zależnościami warunkowymi.

Formalnie, sieć Bayesa to uporządkowana trójka [122]:
$B =< N, E, CP >$,
gdzie:
N – zbiór wierzchołków (węzłów) grafu (ang. *nodes*),
E – zbiór krawędzi grafu (ang. *edges*),
CP – zbiór prawdopodobieństw warunkowych
(ang. *conditional probabilities*).

Wierzchołkami grafu są stwierdzenia lub badane hipotezy. Przy każdym wierzchołku w grafie występuje tabela prawdopodobieństw warunkowych przejścia od danego wierzchołka do następnika. Krawędzie grafu reprezentują prawdopodobieństwa warunkowe zachodzące pomiędzy połączonymi wierzchołkami grafu. Wierzchołki niepołączone są traktowane jako warunkowo niezależne. Łączny rozkład prawdopodobieństwa uzyskujemy jako złożenie warunkowych prawdopodobieństw poszczególnych zmiennych względem ich poprzedników.

Dla przedstawionej w rozdziale 4.2 na stronie 53 wiedzy można stworzyć graf sieci Bayesa widoczny na rysunku 4.1.

Jak widać, największym problemem dla sieci Bayesa jest konieczność określenia wszystkich prawdopodobieństw warunkowych. W przedstawionym przykładzie oszacowano je zgodnie z tabelą 4.1 na stronie 52. Ponadto, należy skorzystać z pomocy eksperta w celu określenia prawdopodobieństwa tego, że kolarz ćwiczy odpowiednio. Jeśli wiedza taka nie będzie dostarczona, wykonany model obarczony będzie dużym ryzykiem błędu. Drugim ze sposobów jest tzw. sampling [133] polegający na zebraniu wielu obserwacji



Rysunek 4.1: Sieć Bayesa dla przykładowej wiedzy

spośród podobnych sytuacji (t.j. kolarzy zawodowych) i na ich podstawie oszacowania jakie jest prawdopodobieństwo, że losowy kolarz dobrze przygotowuje się do treningu.

Dla celów przykładu, zakłada się, że Jan Kolaż trenuje zawodowo i z wartością prawdopodobieństwa równą 0,9 trenuje odpowiednio.

Widocznym problemem jest również konieczność sumowania się prawdopodobieństw do wartości 1. W przedstawionym przykładzie mamy do czynienia z przekroczeniem tej wartości. Jednym ze sposobów na poradzenie sobie z tą sytuacją, jest przeskalowanie wszystkich wartości tak, aby ich suma dawała wartość prawdopodobieństwa zdarzenia pewnego.

Formalnie:

$B =< N, E, CP >$
Gdzie:
$N =$ {stan roweru (S), intensywność treningu (T), wypadek na trasie (W), wynik maratonu (M), premia(P), humor sponsora (H)}

$E = \{$stan roweru $\Rightarrow$ wynik maratonu, intensywność treningu $\Rightarrow$ wynik maratonu, wypadek na trasie $\Rightarrow$ wynik maratonu, wynik maratonu $\Rightarrow$ premia, humor sponsora $\Rightarrow$ premia$\}$
$CP = \{P(S_{dobry}) = 0,9;\ P(S_{zly}) = 0,1;\ P(T_{odpowiedni}) = 0,9;\ P(T_{nieodpowiedni}) = 0,1;\ P(W_{Tak}) = 0,05;\ P(W_{Nie}) = 0,95;\ P(H_{dobry}) = 0,6;\ P(H_{zly}) = 0,4;\ \}$

Z racji dużej złożoności, pozostałe prawdopodobieństwa przedstawione są na rysunkach 4.2 oraz 4.3.

Stan roweru	Dobry				Zły			
Intensywnosc ...	Odpowiednia		Nieodpowiednia		Odpowiednia		Nieodpowiednia	
Wypadek na tr...	Tak	Nie	Tak	Nie	Tak	Nie	Tak	Nie
Zwyciestwo	0.6	0.8	0	0.4	0.0769230...	0.23076923	0	0.2
Podium	0.2	0.1	0	0.4	0.11538462	0.63846154	0.05	0.73
Przegrana	0.2	0.1	1	0.2	0.80769231	0.13076923	0.95	0.07

Rysunek 4.2: Prawdopodobieństwa warunkowe dane w postaci tabeli

Wynik maratonu	Zwyciestwo		Podium		Przegrana	
Humor sponsora	Dobry	Zły	Dobry	Zły	Dobry	Zły
Przyznana	1	0.7	1	0.5	0	0
Nieprzyznana	0	0.3	0	0.5	1	1

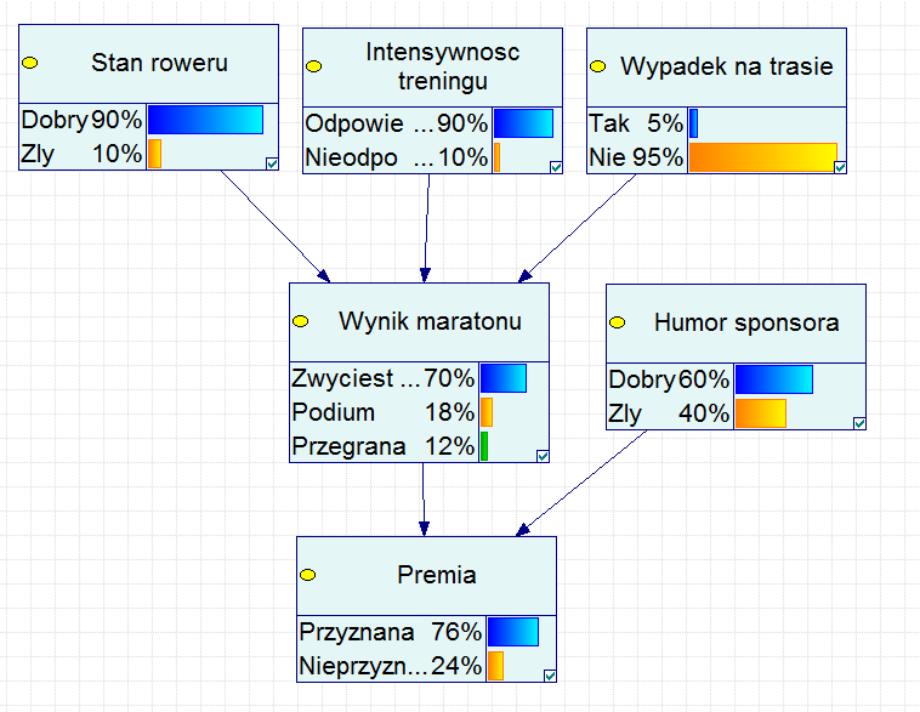
Rysunek 4.3: Prawdopodobieństwa warunkowe dane w postaci tabeli

Wnioskowanie w sieci Bayesa sprowadza się do obliczenia sumy prawdopodobieństw warunkowych zgodnie z połączeniami w tej sieci. Obliczenia te mogą zostać wykonane automatycznie, a ich wynik dany jest na rysunku 4.4. Jak widać, z wartością prawdopodobieństwa równą 0,76 Janek otrzyma premię na zawodach.

Przykładowa sieć została wykonana w programie GeNiE[†]. Automatycznie odzwierciedlone i wyliczone są już wartości poszczególnych prawdopodobieństw łącznych i warunkowych.

Istnieją efektywne algorytmy wnioskujące w sieciach Bayesa. Niestety, ogromną wadą tychże sieci jest brak skalowalności w przypadku zastosowań o dużej złożoności. W celu poprawnego zamodelowania pełnej sieci Bayesa konieczne jest zdefiniowanie $2^m - 1$ parametrów, gdzie m to liczba węzłów w sieci.

[†] http://genie.sis.pitt.edu/



Rysunek 4.4: Wnioskowanie w sieci Bayesa

Pewną dodatkową niedogodnością jest fakt, iż wartości prawdopodobieństw muszą się sumować do jedynki. Otóż, w przypadku, gdy za pomocą teorii prawdopodobieństwa modelujemy wybrany fragment rzeczywistości (często bardzo złożony), nie możemy się ograniczać do logiki dwuwartościowej i prawa *"tertium non datur"* tłumaczonego jako trzeciego wyjścia nie ma. Czasami ekspert nie może przewidzieć prawdopodobieństwa jednego ze zdarzeń bez powiązania go z innymi zdarzeniami warunkującymi. W szczególnym przypadku, zdarzenie może zajść gdy chociaż jedno (lub wszystkie) zdarzenie warunkujące będą spełnione. Między innymi te powody oraz brak możliwości zakładania niezależności zdarzeń wchodzących w skład sieci Bayesa pociąga za sobą fakt trudności w ich stosowaniu.

Akwizycja wiedzy do wykorzystania w modelach probabilistycznych również jest utrudniona ze względu na to, że dla ekspertów dziedzinowych (np. z zakresu medycyny) posługiwanie się statystyką i rachunkiem prawdopodobieństwa jest problematyczne; nie potrafią oni określać wartości prawdopodobieństwa w sposób poprawny.

Sieci bayesowskie mogą być konstruowane, nawet jeśli tylko część właściwości warunkowej niezależności zmiennych jest znana. Istotną zaletą jest

także i to, że taką sieć można zbudować mając niepełne dane na temat zależności warunkowej zdarzeń oraz fakt, że sieci bayesowskie potrafią po prostu bardziej zwarcie reprezentować rozkład prawdopodobieństwa.

Wnioskowanie w sieciach Bayesa może przebiegać w dwojaki sposób. Pierwszy z nich określany mianem predykcyjnego stanowi podejście typu "od góry do dołu" [15]. Polega on na analizie prawdopodobieństw warunkowych i całkowitych wszystkich węzłów będących przodkami rozpatrywanego węzła. Korzystając z własności określonych na wstępie tego rozdziału możliwe jest określenie wartości prawdopodobieństwa całkowitego rozpatrywanego węzła.

Drugi rodzaj wnioskowania to wnioskowanie diagnostyczne, inaczej zwane "od dołu do góry" [15]. Tu analizowane są wartości dowodów (hipotez) będących potomkami rozpatrywanego węzła. Przykładowo, mając daną sieć Bayesa w której mamy określone symptomy choroby (np. kaszel, katar, wysoka temperatura ciała) połączone z konkluzjami będącymi faktycznymi jednostkami chorobowymi możemy dokonać wnioskowania. Wiedząc na jakie choroby pacjent choruje (np. wykonując inne testy) można sprawdzić jakie jest prawdopodobieństwo występowania rozpatrywanych symptomów. Z drugiej strony – znając symptomy choroby jesteśmy w stanie sprawdzić prawdopodobieństwo wystąpienia danej choroby u pacjenta.

Należy tutaj zauważyć, że nawet dla sieci Bayesa, w której wszystkie węzły są binarne (czyli mają określone prawdopodobieństwo dokładnie jednego zdarzenia i zdarzenia mu przeciwnego), złożoność wnioskowania dokładnego jest określona jako $O(2^n)$ [15]. Daje to wykładniczy czas, a co za tym idzie klasyfikuje algorytm do klasy problemów NP-trudnych. Istnieją wprawdzie algorytmy zmniejszające tą złożoność, ale działają one tylko dla specyficznych podklas problemu [86, 123].

4.4 Teoria Dempstera-Shafera

Pewną odpowiedzią i kontr-teorią dla teorii prawdopodobieństwa i sieci Bayesa była teoria opracowana w roku 1960 przez Arthura Dempstera [167]. W późniejszym czasie Glenn Shafer dokonał jej ulepszeń, stąd dziś mówi się o teorii Depmstera-Shafera (DS, zwaną również teorią funkcji przekonania – Matematyczną Teorią Ewidencji [MTE]) [137].

Jednym z największych problemów teorii prawdopodobieństwa, utrudniającym wykorzystanie jej w kontekście systemów ekspertowych, jest konieczność podania wartości wszystkich prawdopodobieństw występujących

w danym modelu. Co więcej, prawdopodobieństwa zdarzeń dotyczących tego samego zjawiska muszą sumować się do wartości 1. Bardzo często jednak zdarza się, że nie są dane wartości prawdopodobieństw niektórych zdarzeń (model jest niepełny) lub też eksperci nie są w stanie dojść do konsensusu aby suma prawdopodobieństw dotyczących tej samej zmiennej była równa 1.

W teorii DS nie jest konieczne spełnienie tych powyższych dwóch warunków. Jest ona stosowana gdy mamy do czynienia z wiedzą niepełną, aktualizacją przekonań i przeprowadzaniem dowodzenia. Uznaje ona model częściowo wyspecjalizowany, w którym nie ma potrzeby uzupełniania brakującej specyfikacji. Kolejną różnicą w porównaniu do teorii Bayesa jest wyznaczanie prawdopodobieństwa możliwości udowodnienia hipotezy na podstawie posiadanej informacji, a nie samych prawdopodobieństw prawdziwości tychże hipotez. Celem teorii Dempstera-Shafera jest rozróżnienie między niepewnością a niewiedzą.

Każdy fakt w teorii DS ma przypisywaną wartość wsparcia (z przedziału [0...1]). Wartość 0 oznacza brak wsparcia dla faktu, 1 – pełne wsparcie.

Wprowadza się również zbiór możliwych konkluzji Θ:

$$\Theta = \{\theta_1, \theta_2, \ldots, \theta_n\}$$

taki, że dokładnie jedna wartość ze zbioru Θ jest prawdziwa. W teorii Bayesa rozpatrywane było prawdopodobieństwo każdego elementu z tego zbioru, teoria Dempstera-Shafera rozpatruje prawdopodobieństwo zajścia zdarzeń będących podzbiorami zbioru możliwych konkluzji.

W teorii DS rozpatrywana jest również przestrzeń stanów [zwana także ramą rozróżniającą (ang. *State space, frame of discernment*)] będącą w istocie zbiorem potęgowym:

$$\text{Jeżeli } \Theta = \{\theta_1, \theta_2, \theta_3\}, \text{ to}$$

$$2^{\Theta} = \{\emptyset, \theta_1, \theta_2, \theta_3, \{\theta_1, \theta_2\}, \{\theta_2, \theta_3\}, \{\theta_1, \theta_3\}, \{\theta_1, \theta_2, \theta_3\}\}$$

Zgodnie z aksjomatami teorii DS:

- $\emptyset$ ma prawdopodobieństwo równe 0, ze względu na to, że przynajmniej jedna konkluzja musi być prawdziwa.
- Pozostałe elementy mają prawdopodobieństwo z przedziału [0...1].

- Prawdopodobieństwo zbioru $\{\theta_1, \theta_2, \theta_3\}$ jest równe 1, ponieważ co najmniej jeden z elementów musi być prawdziwy.

Funkcję m przyporządkowującą prawdopodobieństwo dla elementów zbioru potęgowego nazywamy podstawowym przyporządkowaniem prawdopodobieństwa dla zbioru Θ:

$$\begin{aligned} m(\theta) &\in [0 \dots 1] \\ m(\emptyset) &= 0 \\ \sum_{\theta \in \Theta} &= 1 \end{aligned}$$

Wreszcie dla każdego $X \subseteq \Theta$, definiuje się nowe miary:

$$Bel(X) = \sum_{Y \subseteq X} m(Y) \tag{4.1}$$

$$Pl(X) = \sum_{Y \cap X \neq \emptyset} m(Y) \tag{4.2}$$

$$Doub(X) = 1 - Pl(X) = Bel(\neg X) \tag{4.3}$$

Równanie 4.1 nazywane jest miarą przekonania (ang. *belief*) i określa ono wiarygodność poszlak przemawiających na korzyść X. Równanie 4.2 to funkcja wiarygodności (ang. *plausibility*) liczbowo określające siłę poszlak przemawiających przeciwko X. Kombinacja tych dwóch równań to miara wątpliwości 4.3 (ang. *doubt*). Miarę $Bel(X)$ można traktować jako dolną granicę prawdopodobieństwa, natomiast $Pl(X)$ jako górną granicę.

Definiowany jest również przedział przekonania (ang. *relief interval*) $[X \in \Theta : \ Bel(X), Pl(X)]$. Jego rozpiętość jest interpretowana jako wielkość wiedzy o prawdziwości danej hipotezy (im większa rozpiętość wartości, tym mniejsza wiedza).

Aby zapisać przykładową wiedzę dziedzinową przedstawioną w rozdziale 4.2 na stronie 53 zgodnie z zasadami Matematycznej Teorii Ewidencji, dokonajmy założenia, że "stan roweru", "intensywność treningu" oraz "wypadek na trasie" to trzy niezależne oszacowania mające swoje oddzielne ramy rozróżniające, a co za tym idzie – trzy podstawowe przyporządkowania prawdopodobieństwa:

Dla Θ_1 (stan roweru):

$$
\begin{aligned}
m(\emptyset) &= 0 \\
m(dobry) &= 0,6 \\
m(zly) &= 0,3 \\
m(dobry \cup zly) &= 0,1
\end{aligned}
$$

Dla Θ_2 (intensywność ćwiczeń):

$$
\begin{aligned}
m(\emptyset) &= 0 \\
m(odpowiednia) &= 0,5 \\
m(nieodpowiednia) &= 0,3 \\
m(odpowiednia \cup nieodpowiednia) &= 0,2
\end{aligned}
$$

Dla Θ_3 (wypadek na trasie):

$$
\begin{aligned}
m(\emptyset) &= 0 \\
m(tak) &= 0,7 \\
m(nie) &= 0,1 \\
m(tak \cup nie) &= 0,2
\end{aligned}
$$

Dla wszystkich zbiorów $\Theta_1, \Theta_2, \Theta_3$ wyliczone zostaną miary przekonania, wiarygodności i wątpliwości (tab. 4.3, 4.4, 4.5).

X	dobry	zły	dobry, zły
$m(X)$	0,6	0,3	0,1
$Bel(X)$	0,6	0,3	1
$Pl(X)$	0,7	0,4	1
$Doub(X)$	0,3	0,6	0

Tabela 4.3: Miary przekonania, wiarygodności i wątpliwości dla Θ_1 (stan roweru)

W podanym przykładzie mamy do czynienia z jednym ekspertem, więc dokonywanie kombinacji Dempstera nie jest wymagane.

X	odpowiedni	nieodpowiedni	odpowiedni, nieodpowiedni
$m(X)$	0,5	0,3	0,2
$Bel(X)$	0,5	0,3	1
$Pl(X)$	0,7	0,5	1
$Doub(X)$	0,3	0,5	0

Tabela 4.4: Miary przekonania, wiarygodności i wątpliwości dla Θ_2 (intensywność ćwiczeń):

X	tak	nie	tak, nie
$m(X)$	0,7	0,1	0,2
$Bel(X)$	0,7	0,1	1
$Pl(X)$	0,9	0,3	1
$Doub(X)$	0,1	0,3	0

Tabela 4.5: Miary przekonania, wiarygodności i wątpliwości dla Θ_3 (wypadek na trasie)

Wyniki obliczeń pokazują, że największa wątpliwość znajduje się w sytuacji, gdy stan roweru jest określany jako zły. Informacja ta może posłużyć do baczniejszego przyglądnięcia się tej sytuacji i jej wpływu na wynik maratonu Janka.

W oparciu o przedstawianą teorię opracowano reprezentację wieloatrybutowych rozkładów przekonań w postaci tzw. łańcuchów Markowa. Zgodnie z definicją, jest to nieskierowane drzewo z węzłami reprezentującymi zbiory atrybutów spełniających pewne warunki sąsiedztwa [105]. Węzły w takim drzewie mają określone wartości składowych funkcji przekonania, a łączny ich rozkład jest zdefiniowany zgodnie z definicją złożenia dempsterowskiego składowych funkcji przekonania omówionego wyżej. Tak stworzona struktura umożliwia wykorzystanie algorytmów wnioskowania progresywnego i regresywnego.

Zastosowanie teorii DS jest niestety utrudnione z podobnego powodu, dla którego sieci Bayesa były trudne w stosowaniu. Nie istnieje dobra metoda akwizycji wiedzy dla modelu DS, ani bezpośrednio z danych ani poprzez jej weryfikację wobec otrzymanych informacji. Empiryczne pozyskiwanie danych jest niemożliwe.

4.5 Teoria zbiorów rozmytych oraz logika rozmyta

Kolejną z metod stosowaną w przypadku wiedzy niepełnej jest zaproponowana przez L. Zadeha w 1965 r. teoria zbiorów rozmytych [170].

Klasyczna logika dwuwartościowa (typu prawda-fałsz) okazywała się niewystarczająca do praktycznych zastosowań. Podwaliny pod nową teorię zaproponował polski uczony Jan Łukasiewicz w swoim traktacie "O zasadzie sprzeczności u Arystotelesa" [156]. Oprócz dwóch znanych uprzednio stanów logicznych, Łukasiewicz dołożył trzeci – "możliwość" o wartości liczbowej $\frac{1}{2}$ reprezentującą wiedzę w postaci "pół na pół". Oprócz tego, zaproponowana była arytmetyka trójwartościowa uwzględniająca ten dodatkowy stan oraz zbiór aksjomatów z całościową notacją. W późniejszym okresie powstała logika wielowartościowa i koncepcja logiki o nieskończonej liczbie wartości.

Drugim polskim matematykiem, którego prace dały początek teorii zbiorów rozmytych, był Stanisław Leśniewski [87]. Stworzona przez niego teoria o nazwie "mereologia" zakładała zmianę aksjomatu przynależności elementu do zbioru. Leśniewski stwierdził, że dany element nie jest elementem, ale częścią zbioru. Stąd blisko już do rozszerzenia tych zasad, podania metod operacji na zbiorach i aksjomatów aż do uzyskania zbiorów rozmytych. Co ciekawe, prace te zostały również wykorzystane przez prof. Zdzisława Pawlaka w omawianej później teorii zbiorów przybliżonych.

Człowiek w sposób bardziej naturalny używa określeń jakościowych do opisu rzeczywistości, np. "wysokie drzewo", "silny mężczyzna", itp. Z nieostrością pojęć wiąże się również subiektywność w odbiorze informacji. Waga ciała równa 90 kg to dla osoby niskiej już otyłość, natomiast kulturysta uzna ją za niedowagę. Owa subiektywność to właśnie swoiste rozmycie granic zbioru obiektów. Zgodnie z klasyczną teorią zbiorów, osoba taka nie może jednocześnie należeć do zbioru osób otyłych i z niedowagą. W celu wyeliminowania tej nieprecyzyjności wiedzy, wprowadza się pojęcie zbioru rozmytego:

Zbiorem rozmytym [77] A w pewnej (niepustej) przestrzeni X, $A \subseteq X$, nazywamy zbiór par

$$A = \{(x, \mu_A(x)); x \in X\}$$

gdzie

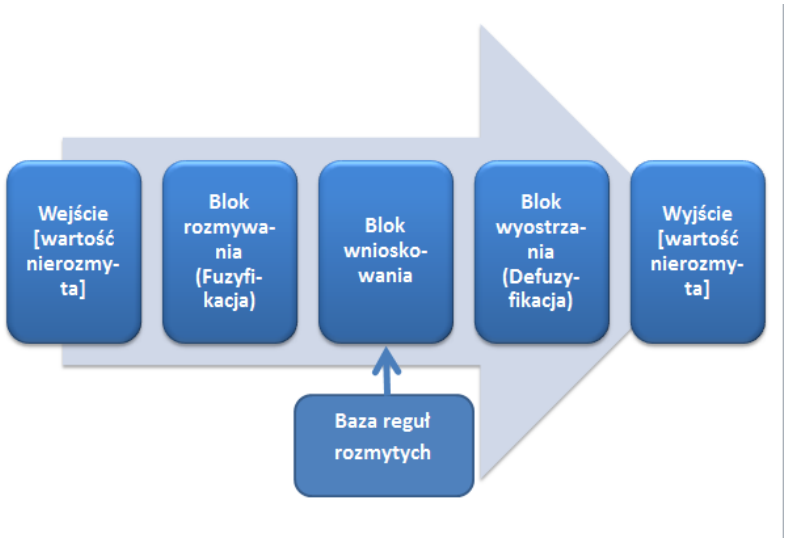
$$\mu_A : X \to [0 \ldots 1]$$

jest funkcją przynależności zbioru rozmytego A.
Właściwości funkcji przynależności są następujące:

- Funkcja każdemu elementowi $x \in X$ przyporządkowuje wartość z przedziału $[0 \ldots 1]$,
- $\mu_A(x) = 1$ oznacza pełną przynależność do zbioru rozmytego,
- $\mu_A(x) = 0$ oznacza brak przynależności do zbioru rozmytego,
- $0 < \mu_A(x) < 1$ oznacza częściową przynależność do zbioru rozmytego.

W pracy [28] znaleźć można wyczerpujący opis topologii, aksjomatów oraz relacji i ich typów, jakie zachodzą pomiędzy zbiorami rozmytymi.

Schemat procesu wnioskowania rozmytego jest przedstawiony na rysunku 4.5:



Rysunek 4.5: Schemat wnioskowania rozmytego

Wejście Na wejściu algorytmu wnioskującego przekazywana jest wartość nierozmyta.

Fuzyfikacja jest operacją zamieniającą sygnały wejściowe z dziedziny ilościowej na wielkości jakościowe reprezentowane przez zbiory rozmyte na podstawie określających je funkcji przynależności.

Wnioskowanie odbywa się z wykorzystaniem bazy reguł rozmytych i polega na przekształcaniu wejściowych wartości zmiennych lingwistycznych do wyjściowych wartości zmiennych lingwistycznych.

Defuzyfikacja jest operacją odwrotną do fuzyfikacji. Wartości zmiennych lingwistycznych zostają przekształcone na wartości nierozmyte. Istnieje wiele metod wyostrzania, najczęściej stosowane to metoda środka maksimum, metoda pierwszego (ostatniego) maksimum oraz metoda środka ciężkości [77].

Wyjście Wyjście algorytmu wnioskującego stanowi wartość nierozmyta.

Istnieje wiele modeli wnioskowania na podstawie rozmytej bazy wiedzy, wśród których można wyróżnić dwie kategorie [100]. Pierwsza z nich, to modele lingwistyczne (np. model typu Mamdami) opierające się na regułach postaci "jeżeli ... to ..." analogicznych do tych używanych w systemach ekspertowych. Różnica polega na tym, że w miejsce zbioru atrybutów użyty jest zbiór zmiennych lingwistycznych, a w miejsce zbioru wartości atrybutów – zbiory wartości zmiennych lingwistycznych.

Druga kategoria zawiera modele oparte na wnioskowaniu Takagi-Sugeno. Tworzone są one przez reguły logiczne z rozmytą częścią warunków oraz funkcyjny następnik. Są *de facto* kombinacją modeli rozmytych i liniowych.

Zapis przykładu dla przedstawionej wcześniej wiedzy dziedzinowej przy użyciu logiki rozmytej wymaga określenia zbiorów rozmytych dla poszczególnych zmiennych i ich wartości. Wymagana jest również synteza wiedzy do postaci reguł rozmytych:

1. JEŻELI stan roweru JEST dobry ORAZ intensywność treningu JEST odpowiednia ORAZ zdarzył się wypadek na trasie JEST tak TO Janek zwycięża.

2. JEŻELI stan roweru JEST dobry ORAZ intensywność treningu JEST odpowiednia ORAZ zdarzył się wypadek na trasie JEST nie TO Janek zwycięża.

3. JEŻELI stan roweru JEST dobry ORAZ intensywność treningu JEST nieodpowiednia ORAZ zdarzył się wypadek na trasie JEST nie JEST tak TO Janek przegrywa.

4. JEŻELI stan roweru JEST dobry ORAZ intensywność treningu JEST nieodpowiednia ORAZ zdarzył się wypadek na trasie JEST nie TO Janek staje na podium.

5. JEŻELI stan roweru JEST zły ORAZ intensywność treningu JEST odpowiednia ORAZ zdarzył się wypadek na trasie JEST tak TO Janek przegrywa.

6. JEŻELI stan roweru JEST zły ORAZ intensywność treningu JEST odpowiednia ORAZ zdarzył się wypadek na trasie JEST nie TO Janek staje na podium.

7. JEŻELI stan roweru JEST zły ORAZ intensywność treningu JEST nieodpowiednia ORAZ zdarzył się wypadek na trasie JEST tak TO Janek przegrywa.

8. JEŻELI stan roweru JEST zły ORAZ intensywność treningu JEST nieodpowiednia ORAZ zdarzył się wypadek na trasie JEST nie TO Janek staje na podium.

9. JEŻELI wynik maratonu JEST zwycięstwo ORAZ humor sponsora JEST dobry TO premia JEST przyznana.

10. JEŻELI wynik maratonu JEST zwycięstwo ORAZ humor sponsora JEST zły TO premia JEST przyznana.

11. JEŻELI wynik maratonu JEST podium ORAZ humor sponsora JEST dobry TO premia JEST przyznana.

12. JEŻELI wynik maratonu JEST podium ORAZ humor sponsora JEST zły TO premia JEST nieprzyznana.

13. JEŻELI wynik maratonu JEST przegrana ORAZ humor sponsora JEST dobry TO premia JEST nieprzyznana.

14. JEŻELI wynik maratonu JEST przegrana ORAZ humor sponsora JEST zły TO premia JEST nieprzyznana.

Przedstawione reguły nie wyczerpują wszystkich możliwych przypadków, jednakże złożoność reguł modelujących całą rzeczywistość zapisaną w tekście przekracza ramy tego przykładu. Należy również zaznaczyć, że określenia "wygrywa", "przegrywa", "staje na podium" są pewnym uproszczeniem. Mamy tu do czynienia z jedną zmienną konkluzyjną o nazwie wynik, a jej wartościami są wyżej przytoczone pojęcia. W pełni poprawny (aczkolwiek mniej intuicyjny) zapis jednej z reguł wyglądałby następująco:

```
JEŻELI stan roweru JEST dobry
ORAZ intensywność treningu JEST odpowiednia
ORAZ zdarzył się wypadek na trasie JEST tak
TO Janek ma wynik JEST wygrywa.
```

Aby przeprowadzić prawidłowy proces wnioskowania należałoby przedstawić wzory do obliczania funkcji przynależności dla każdej wartości zmiennych występujących w powyższych regułach. Należy również zaproponować metodę określania zmiennych jakościowych takich jak "stan roweru" wartościami liczbowymi możliwymi do zastosowania jako argument funkcji przynależności. Załóżmy więc, że stan roweru oraz intensywność treningu będziemy określać jako wartość z przedziału $[0 \ldots 10]$. W ten sposób zamodelować można naturalne zachowanie człowieka, znane choćby z badań lekarskich i intensywności odczuwania bólu. Funkcje przynależności dla tych zmiennych lingwistycznych kształtują się więc następująco:

$$\mu_{stan\ roweru\ dobry}(x) = \begin{cases} 0 \text{ dla } x \leqslant 3 \\ \frac{x-3}{5} \text{ dla } 3 < x \leqslant 8 \\ 1 \text{ dla } x > 8 \end{cases}$$

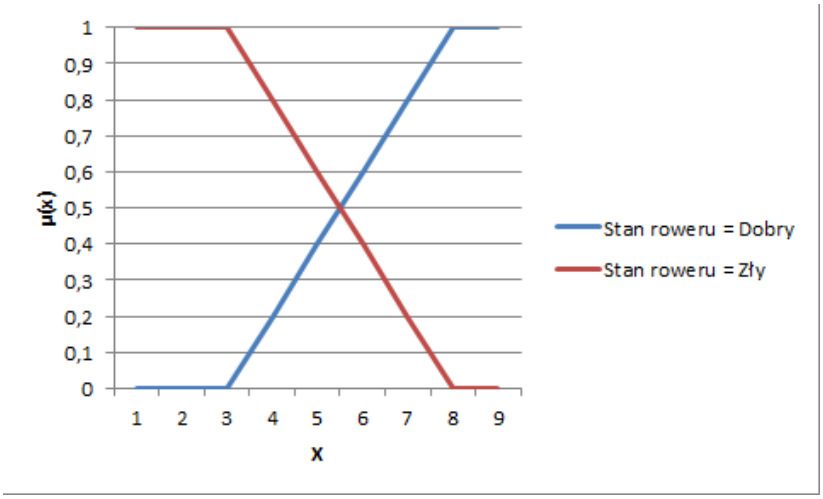
$$\mu_{stan\ roweru\ zly}(x) = \begin{cases} 1 \text{ dla } x \leqslant 3 \\ \frac{8-x}{5} \text{ dla } 3 < x \leqslant 8 \\ 0 \text{ dla } x > 8 \end{cases}$$

$$\mu_{trening\ odpowiedni}(x) = \begin{cases} 0 \text{ dla } x \leqslant 3 \\ \frac{x-3}{5} \text{ dla } 3 < x \leqslant 8 \\ 1 \text{ dla } x > 8 \end{cases}$$

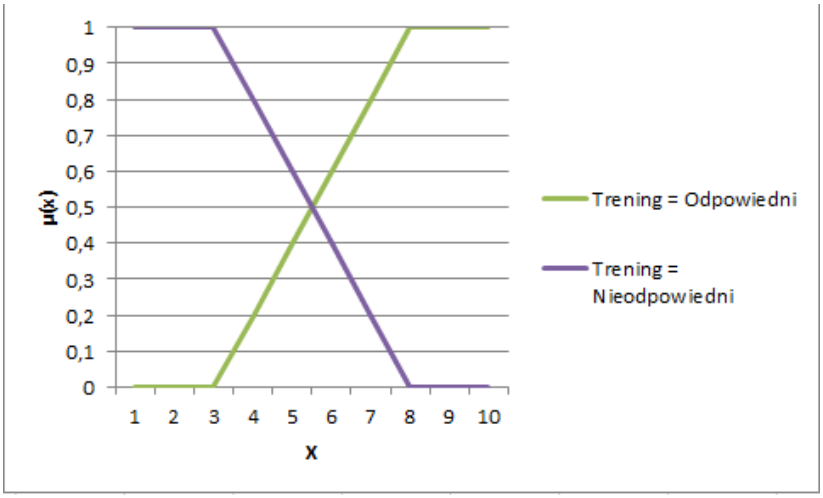
$$\mu_{trening\ nieodpowiedni}(x) = \begin{cases} 1 \text{ dla } x \leqslant 3 \\ \frac{8-x}{5} \text{ dla } 3 < x \leqslant 8 \\ 0 \text{ dla } x > 8 \end{cases}$$

Wykresy powyższych funkcji przynależności obrazują rysunki 4.6 oraz 4.7 na stronie 69.

Funkcję przynależności dla zmiennej "zdarzył się wypadek" określmy jako funkcję stałą, przyjmującą wartości 1 oraz 0 dla wartości "tak" oraz "nie" odpowiednio. I znów, dokonujemy tu pewnego rodzaju uproszczenia w stosunku do modelu rzeczywistego, który powinien zbadać wpływ wypadku na pozostałe czynniki uwzględnione wcześniej.



Rysunek 4.6: Wykres funkcji przynależności zmiennej lingwistycznej stan roweru



Rysunek 4.7: Wykres funkcji przynależności zmiennej lingwistycznej trening

Załóżmy więc, że stan roweru został określony przez Janka wartością 4, a lekarz ocenił, że Janek trenował w 90% (wartość 9) tak jak powinien. Zgodnie z określonymi wcześniej funkcjami przynależności mamy:

$$\mu_{stan\ roweru\ dobry}(4) = 0,2$$

$$\mu_{stan\ roweru\ zly}(4) = 0,8$$

$$\mu_{trening\ odpowiedni}(9) = 1$$

$$\mu_{trening\ nieodpowiedni}(9) = 0$$

Zakładając również, że nie zdarzył się wypadek na trasie, jedynymi regułami dotyczącymi tego, czy Janek zwycięży możliwymi do uaktywnienia (posiadającymi wszystkie wartości funkcji przynależności konkluzji większe od 0) będą reguły:

1. JEŻELI stan roweru JEST dobry ORAZ intensywność treningu JEST odpowiednia ORAZ zdarzył się wypadek na trasie JEST nie TO Janek zwycięża.

2. JEŻELI stan roweru JEST zły ORAZ intensywność treningu JEST odpowiednia ORAZ zdarzył się wypadek na trasie JEST nie TO Janek staje na podium.

W każdej z reguł znajdują się trzy przesłanki połączone spójnikiem ORAZ. Zgodnie z regułami wnioskowania typu Mamdani w systemach rozmytych, wartość funkcji przynależności zmiennej lingwistycznej będącej konkluzją tychże reguł będzie ograniczona do minimalnej wartości którejkolwiek przesłanki, a więc:

$$\begin{aligned}
&\text{Reg. 1:}\\
&min\{\mu_{stan\ roweru}(dobry);\\
&\mu_{intensywnosc\ treningu}(odpowiednia);\\
&\mu_{wypadek}(nie)\}\\
&= \min\{0,2; 1; 1\} = 0,2\\
&\text{Reg. 2:}\\
&min\{\mu_{stan\ roweru}(zly);\\
&\mu_{intensywnosc\ treningu}(odpowiednia);\\
&\mu_{wypadek}(nie)\}\\
&= \min\{0,8; 1; 1\} = 0,8
\end{aligned}$$

Aby otrzymać ostateczny wynik i informację o spodziewanym zajętym miejscu przez zawodnika, należy również określić funkcję przynależności dla zbioru rozmytego "wynik". Z racji tego, że mamy tu do czynienia z funkcją

operującą na wartościach dyskretnych (nie da się zająć 3,5 miejsca), tym razem funkcja przynależności zadana jest tabelą 4.6.

μ_{wynik}	Wygrana	Podium	Przegrana
1	0,2	0,8	0
2	0,2	0,8	0,8
3	0,2	0,8	0,6
4	0,2	0,8	0,4
5	0,2	0,6	0,2
6	0,2	0,4	0
7	0,2	0,2	0
8	0,2	0	0
9	0,2	0	0
10	0,1	0	0

Tabela 4.6: Funkcja przynależności dla zbioru rozmytego wynik

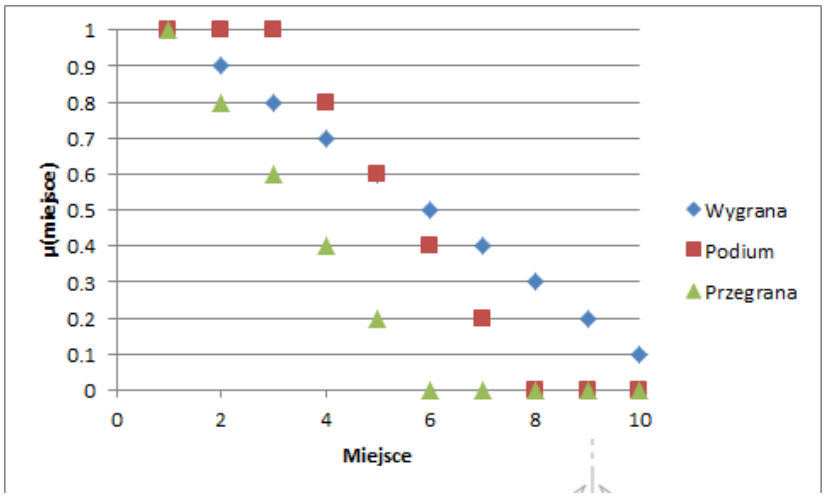
Po ograniczeniu wartości funkcji do wartości odpowiednich wyliczonym wcześniej otrzymujemy wartości zapisane w tabeli 4.7.

μ_{wynik}	Wygrana	Podium	Przegrana
1	0,2	0,8	0
2	0,2	0,8	0,8
3	0,2	0,8	0,6
4	0,2	0,8	0,4
5	0,2	0,6	0,2
6	0,2	0,4	0
7	0,2	0,2	0
8	0,2	0	0
9	0,2	0	0
10	0,1	0	0

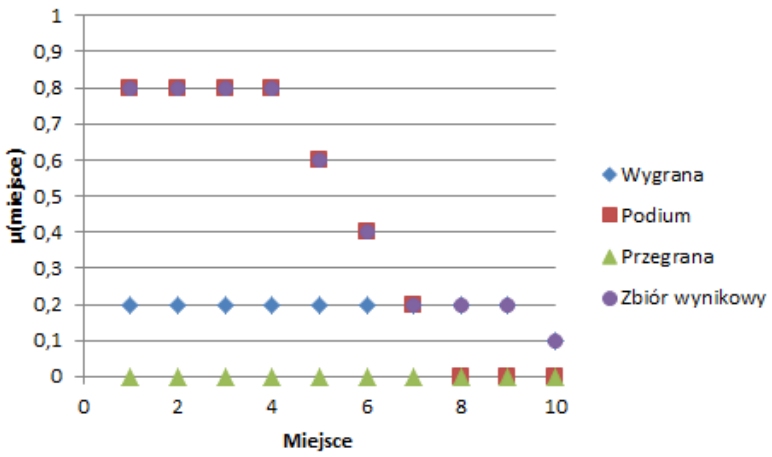
Tabela 4.7: Funkcja przynależności dla zbioru rozmytego wynik po ograniczeniu

Graficzna interpretacja uzyskanych wyników została zilustrowana na rysunkach 4.8 oraz 4.9.

Stosując metodę ostatniego maksimum, otrzymujemy informację, że Janek zajął drugie miejsce. Aby dokończyć wnioskowanie i sprawdzić jaka premia należy się zawodnikowi, należałoby przeprowadzić analogiczny proces dla zmiennych lingwistycznych "humor sponsora" oraz "premia". Ze względu jednak na obszerność tego przykładu i analogię rozumowania, nie będzie on tutaj przytoczony.



Rysunek 4.8: Wykres funkcji przynależności zmiennej lingwistycznej wynik



Rysunek 4.9: Wykres funkcji przynależności zmiennej lingwistycznej wynik po ograniczeniu

Systemy korzystające z logiki rozmytej są powszechnie wykorzystywane w praktyce. Jak napisano wcześniej w rozdziale 2.1.1 na stronie 12, systemy rozmyte zdobywają coraz większą popularność ze względu na szerokie możliwości zastosowania. Do klasycznych zastosowań można zaliczyć sterowanie wszelkimi urządzeniami wyposażonymi w automatykę (od sterowników przemysłowych do sprzętu gospodarstwa domowego), układami sterowania maszynami ciężkimi, reaktorami, czy klimatyzatorami.

W obrębie zastosowań medycznych, systemy rozmyte używane są do

sterowania pompami insulinowymi czy wraz z metodami analizy obrazów - rozpoznawaniem gestów języka migowego.

Pomimo zalet metod korzystających z logiki rozmytej, dużym problemem jest wyznaczanie funkcji przynależności. Jest to proces dość żmudny i trudny, a poprawne rezultaty są trudne do uzyskania.

4.6 Teoria zbiorów przybliżonych

Profesor Zdzisław Pawlak w 1982 r. zaproponował tą nowatorską metodę z powodzeniem wykorzystywaną dzisiaj w modelowaniu zarówno wiedzy pewnej, jak i niepewnej [116].

Wiedza pewna zapisywana jest w sposób omówiony wcześniej (rozdział 2.3.4 na stronie 26).

Proponuje się wykorzystanie pojęć ostrych tzw. *dolnego* i *górnego przybliżenia* w miejsce tych nieprecyzyjnych. Zdefiniowana również różnica pomiędzy dolnym i górnym przybliżeniem (inaczej brzeg zbioru) będzie stanowiła o niepewności wiedzy.

Formalnie, jeśli dany jest system informacyjny $SI = \{U, A, V, f\}$ [117] taki, że:

- U to uniwersum, czyli zbiór obiektów wchodzących w skład systemu,
- A to zbiór atrybutów opisujących obiekty definiowane w systemie,
- V to zbiór wartości atrybutów,
- f to zdefiniowana wcześniej funkcja informacyjna,

można zdefiniować tzw. relację nierozróżnialności będącą iloczynem kartezjańskim na zbiorze obiektów, takim że wartości odpowiednich atrybutów w obu obiektach są identyczne:

$$IND(B) = \{(x, y) \in U \times U : \forall_{A \in B} a(x) = a(y)\}$$

Relacja ta jest zwrotna, symetryczna i przechodnia. Na podstawie podzbioru atrybutów B przypisuje ona do tego samego zbioru obiekty z uniwersum, które mają takie same wartości odpowiednich atrybutów wchodzących w skład zbioru B.

Wreszcie, zdefiniować można pojęcia przybliżeń:

Dolnym przybliżeniem jest to zbiór takich wszystkich obiektów należących do klas relacji nierozróżnialności nad zbiorem B, które w całości zawierają się w klasie decyzyjnej X. Innymi słowy, jest to obszar, w którym zdefiniowane są wszystkie te obiekty, co do których nie ma wątpliwości, że należą one do zbioru tych pojęć w świetle posiadanej wiedzy:

$$\underline{BX} = \cup_{x \in X}\{Y \in IND(B) : Y \subseteq X\}$$

Górne przybliżenie jest to zbiór takich wszystkich obiektów należących do klas relacji nierozróżnialności nad zbiorem B, które mają niepustą część wspólną z klasą decyzyjną X. Są to więc obiekty, które być może należą do takiego zbioru, albowiem nie mamy całkowitej pewności, że do niego nie należą:

$$\overline{BX} = \cup_{x \in X}\{Y \in IND(B) : Y \cap X \neq \emptyset\}$$

Brzeg zbioru to te elementy, co do których nie wiadomo, czy są reprezentantami danego zbioru:

$$BN_B(X) = \overline{B}X - \underline{B}X$$

Nawiążmy teraz do omówionego przykładu dla ilustracji ww. pojęć. Posłużymy się wiedzą dziedzinową zdefiniowaną w rozdziale 4.2 na stronie 53. Ponadto, korzystając ze zdefiniowanych już w czasie wnioskowania rozmytego reguł (rozdział 4.5 na stronie 66) zapiszemy je zgodnie z teorią zbiorów przybliżonych. Zastosujmy zapis w postaci tablicy decyzyjnej (patrz 2.3.4 na stronie 26). Ze względu na fakt wnioskowania dwupoziomowego, koniecznym jest zdefiniowanie dwóch tablic decyzyjnych.

$$DT_1 = (U_1, A_1 \cup \{d_1\})$$

- $U_1 = \{X_1, X_2, \ldots, X_8\}$
- $A_1 = \{$stan roweru (S), intensywność treningu (I), wypadek na trasie (W)$\}$
- $d_1 \notin A_1 =$ wynik maratonu (M),

$$DT_2 = (U_2, A_2 \cup \{d_2\})$$

- $U_2 = \{R_1, R_2, \ldots, R_x\}$
- $A_2 = \{$wynik maratonu (M), humor sponsora (H)$\}$
- $d_2 \notin A_2 =$ premia (P),

X	Stan roweru (S)	Intensywność treningu (I)	Wypadek na trasie (W)	Wynik maratonu (M)
X_1	dobry	odpowiednia	tak	zwycięstwo
X_2	dobry	odpowiednia	nie	zwycięstwo
X_3	dobry	nieodpowiednia	tak	przegrana
X_4	dobry	nieodpowiednia	nie	podium
X_5	zły	odpowiednia	tak	przegrana
X_6	zły	odpowiednia	nie	podium
X_7	zły	nieodpowiednia	tak	przegrana
X_8	zły	nieodpowiednia	nie	podium

Tabela 4.8: Tablica decyzyjna dla przykładowej wiedzy

X	Wynik maratonu (M)	Humor sponsora (H)	Premia (P)
R_1	wygrana	dobry	przyznana
R_2	wygrana	zły	przyznana
R_3	podium	dobry	przyznana
R_4	podium	zły	nie przyznana
R_5	przegrana	dobry	nie przyznana
R_6	przegrana	zły	nie przyznana

Tabela 4.9: Druga tablica decyzyjna dla przykładowej wiedzy

W celu obliczenia górnego i dolnego przybliżenia zbiorów, przedstawmy najpierw w jaki sposób relacja nierozróżnialności dzieli zbiór obiektów względem wartości decyzji:

$$\begin{aligned} X &= X_a \cup X_b \cup X_c \\ X_a &= \{X_1, X_2\}, \\ X_b &= \{X_3, X_5, X_7\}, \\ X_c &= \{X_4, X_6, X_8\} \end{aligned}$$

Zakładamy, że nie wiemy jaki był stan roweru (S) przed wyruszeniem na trasę maratonu. Obliczmy więc brzeg, dolne oraz górne przybliżenie dla pierwszej tabeli decyzyjnej dla zbioru atrybutów warunkowych B={I, W} względem wartości decyzji:

$$
\begin{aligned}
B &= \{I, W\} \\
U/IND_{SI,B} &= \{\{X_1, X_5\}, \{X_2, X_6\}, \{X_3, X_7\}, \{X_4, X_8\}\} \\
\underline{B}X_a &= \emptyset \\
\underline{B}X_b &= \emptyset \\
\underline{B}X_c &= \emptyset \\
\overline{B}X_a &= \{X_1, X_2, X_5, X_6\} \\
\overline{B}X_b &= \{X_1, X_3, X_5, X_7\} \\
\overline{B}X_c &= \{X_2, X_4, X_6, X_8\} \\
BN_B(X_a) &= \{X_1, X_2, X_5, X_6\} \\
BN_B(X_b) &= \{X_1, X_3, X_5, X_7\} \\
BN_B(X_c) &= \{X_2, X_4, X_6, X_8\}
\end{aligned}
$$

Jak widać, wszystkie pojęcia należą do zbioru brzegowego. Oznacza to, że mając dane tylko te informacje, co na wstępie - nie jesteśmy w stanie jednoznacznie powiedzieć, czy Janek wygra, czy przegra. Z racji tegoż faktu, dalsze wnioskowanie bez dookreślenia nowej wiedzy jest bezcelowe.

Teoria zbiorów przybliżonych jest użyteczna w kontekście indukcji reguł z niekompletnych zbiorów danych. Korzystając z tego podejścia można rozróżnić pomiędzy trzema typami niekompletności wiedzy: (1) wartościami utraconymi (danymi zebranymi, ale aktualnie niedostępnymi), (2) wartościami, które łatwo można uzupełnić za pomocą innej wartości ze zbioru danych oraz (3) niekompletnością niebraną pod uwagę (nieważną).

Istnieje wiele metod stosowanych w przypadku niekompletności danych. Gdy niekompletność dotyczy brakującej wartości danego atrybutu, jedną z często stosowanych technik jest uzupełnienie wartości atrybutu. Oczywiście, najbardziej trywialnym podejściem jest w przypadku atrybutów numerycznych uzupełnić braki wartością średnią, zaś w przypadku cech nominalnych – wartością najczęściej występującą (dominantą). Ciekawe podejście przedstawia praca [126], w której braki są zastępowane wartością "*". Ponadto, autor [51] proponuje poszukiwanie zbioru wartości atrybutu w celu znalezienia najbardziej pasującej wartości (ang. *closest fit*).

Pierwszy przypadek niekompletności wiedzy był rozpatrywany przez autorów publikacji [149], natomiast trzecia sytuacja zainteresowała autorkę pracy [83]. W drugiej koncepcji, brakująca wartość atrybutu może być zastąpiona przez dowolną wartość spośród wartości właściwych dla rozpatrywanej dziedziny i właściwych dla tego "typu" obiektów [49]. Przykładowo, jeśli przystępujemy do umowy ubezpieczenia mieszkania, a z jakiegoś powodu aktualna wartość nieruchomości nie jest znana, korzystając z teorii zbiorów przybliżonych można sięgnąć do wartości nieruchomości o podobnym standardzie, metrażu i lokalizacji aby ekstrapolować brakującą wartość.

Dzięki powyższym metodom, zostały opracowane rozwiązania pozwalające na generowanie reguł przy wykorzystaniu danych niekompletnych.

Omówione tutaj metody znalazły również zastosowanie w wielu systemach hybrydowych w dziedzinie uczenia maszynowego i eksploracji danych. Są szczególnie użyteczne do indukowania reguł decyzyjnych i wyboru cech znaczących służących zmniejszeniu wymiarowości rozpatrywanego problemu. Dziedzina wykorzystywania zbiorów przybliżonych obejmuje takie obszary jak bioinformatyka, ekonomia, finanse, medycyna, multimedia, przetwarzanie sygnałów, robotyka oraz ekstrakcja danych z tekstu.

Szersze i bardziej wnikliwe informacje dotyczące teorii zbiorów przybliżonych znacznie wychodziłoby poza ramy tej rozprawy. Wyczerpujące informacje o tej teorii znaleźć można w opracowaniu [79].

4.7 Współczynniki pewności CF

Jak wykazano we wcześniejszym rozdziale, model probabilistyczny do zapisu wiedzy niepewnej, okazał się niewystarczający. Poczynione założenia o niezależności warunkowej zmiennych oraz sumowaniu wartości prawdopodobieństw dotyczących konkretnej zmiennej do jedynki okazały się przydatne w kontekście praktycznym, jednakże bardzo często niedokładnym.

W 1975 r. autorzy [139] przedstawili system MYCIN wraz z modelem tzw. współczynników pewności (ang. *certainty-factor*). Podejście to było nowatorskie jak na tamte czasy i pomimo upływu wielu lat – nadal pozostaje szeroko wykorzystywane w praktyce.

Dwa lata wcześniej, dzięki Richardowi Swinburne'owi [152] przedstawiona została nowa metoda klasyfikacji teorii statystycznych. Konkluzją pracy była obserwacja intuicyjnie znana już wcześniej – klasyczny rachunek prawdopodobieństwa nie nadaje się do zastosowania w systemach ekspertowych bazujących w większości na cechach jakościowych, a nie ilościowych.

W systemie Mycin skorzystano zatem z możliwości sprawdzania wiarygodności hipotez czy prawdziwości przypuszczeń (domniemań) zdefiniowanych za sprawą logicznej teorii prawdopodobieństwa (ang. *Logical Theory*). Teoria ta głosiła istnienie logicznej relacji pomiędzy obserwacją e a hipotezą h. Wyróżniano tutaj stopień potwierdzenia hipotezy przez zaistniałe przesłanki [20].

Hipotezę h można potwierdzić na jeden z poniższych sposobów:

1. Metodą klasyfikacji, gdzie po prostu obserwacja e potwierdza hipotezę h.

2. Metodą porównywania, gdzie porównywane są co najmniej dwie obserwacje e_1 oraz e_2 i szacowany jest wpływ każdej z nich na potwierdzenie hipotezy h. W wyniku czego przykładowo dochodzimy do wniosku, że "obserwacja e_2 mocniej niż e_1 wpływa na potwierdzenie hipotezy h.

3. Metodą ilościową, gdzie podawana jest wartość liczbowa siły potwierdzania hipotezy.

W systemie Mycin zastosowany został ostatni sposób oceny, co zaowocowało powstaniem miary współczynników pewności CF.

System Mycin zawierał początkowo bazę około 200 reguł w postaci "jeżeli ... to ...". Przykładowa reguła wyglądała następująco:

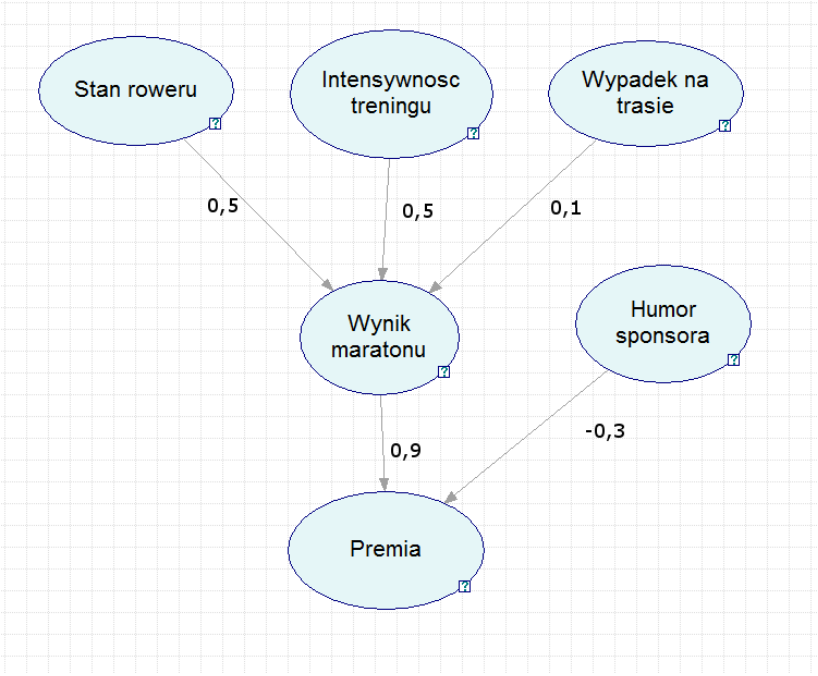
```
IF Kultura bakteryjna rozwinęła się we krwi
AND odczyn jest gramopozytywny
AND bakterie wniknęły przez jelito i żołądek
OR miednica jest miejscem infekcji
THEN Istnieją silne poszlaki (0.7), że klasą bakterii,
które są za toodpowiedzialne jest Enterobacteriacae.
```

Różnica wobec klasycznego zapisu reguły ujawnia się w jej konkluzji, która to jest opisana wartością liczbową współczynnika CF określającego przekonanie o prawdziwości danej reguły. Reguły takie tworzą sieć zależności zwaną siecią współczynników CF. Przykładowa sieć widoczna jest na rysunku 4.10. Zapis wiedzy z wykorzystaniem współczynników pewności CF przedstawiony zostaje analogicznie w stosunku do podejścia probabilistycznego w sieciach Bayesa.

W przypadku naszego przykładu, korzystającego z wiedzy dziedzinowej zdefiniowanej w rozdziale 4.2 na stronie 53, mamy dany zbiór węzłów (N) i krawędzi (E):

N = {stan roweru (S), intensywność treningu (T), wypadek na trasie (W), wynik maratonu (M), premia(P), humor sponsora (H)}
E = {stan roweru ⇒ wynik maratonu, intensywność treningu ⇒ wynik maratonu, wypadek na trasie ⇒ wynik maratonu, wynik maratonu ⇒ premia, humor sponsora ⇒ premia}

Model ten znacznie upraszcza budowę samej sieci, albowiem liczba współczynników, jaką należy podać, jest znacznie mniejsza. Nie są tu już potrzebne prawdopodobieństwa warunkowe wszystkich ewentualności, lecz dla każdego przejścia – dokładnie jedna wartość współczynnika CF. Po odpytaniu eksperta o wpływ poszczególnych czynników na ostateczny wynik, skonstruowany został graf współczynników CF jak na rysunku 4.10.



Rysunek 4.10: Sieć współczynników CF dla przykładowej wiedzy

Na rysunku 4.10 widać przesłanki `Stan roweru`, `Intensywność treningu`, `Wypadek na trasie`, `Wynik maratonu`, `Humor sponsora`, hipotezę `Premia` oraz wartości współczynników CF pomiędzy poszczególnymi przesłankami i hipotezą. Sieć konstruowana jest na podstawie zbioru reguł wyindukowanych z wiedzy dziedzinowej przedstawionej w rozdziale 4.2 na stronie 53.

Jak widać, współczynnik CF przyjmuje wartości z przedziału $[-1; 1]$, jednakże nie jest to wymagane i jego wartość mogłaby być zdefiniowana na innym przedziale. Aby zdefiniować formalnie czym jest CF należy uprzednio dokonać opisu miar z których korzysta się do jego obliczania. W poniższych oznaczeniach przyjmuje się, że $P(h)$ to prawdopodobieństwo zajścia hipotezy h, natomiast $P(h|e)$ to prawdopodobieństwo zajścia hipotezy h pod warunkiem zajścia warunku e.

Miara wiarygodności (zaufania) (ang. *MB – measure of belief*) będącą współczynnikiem przyjmującym wartości z przedziału $[0; 1]$ i określającym stopień przekonania o słuszności danej przesłanki jest definiowana jako:

$$MB[h, e] = \frac{P(h|e) - P(h)}{1 - P(h)}$$

Miara wątpliwości (ang. *MD – measure of disbelief*) będąca współczynnikiem przyjmującym wartości z przedziału $[0; 1]$ i określającymi stopień przekonania o fałszu danej przesłanki jest przedstawiona wzorem:

$$MD[h, e] = \frac{P(h) - P(h|e)}{P(h)}$$

Jeśli $P(h) > P(h|e)$ przyjmuje się, że poznanie wartości obserwacji e zmniejsza przekonanie eksperta o słuszności hipotezy h. W przeciwnym przypadku, słuszność ta zostaje zwiększona dzięki otrzymanej nowej informacji.

Współczynnik pewności CF jest definiowany jako różnica pomiędzy miarą wiarygodności, a wątpliwości:

$$CF[h, e] = MB[h, e] - MD[h, e]$$

Wartość współczynnika CF jest zwykle określana na przedziale $[-1 \ldots 1]$ właśnie ze względu na wartość współczynników $MB[h, e]$ oraz $MD[h, e]$:

- Gdy $CF[h, e] = -1$, oznacza to, że h jest fałszywe na pewno, bo $MB[h, e] = 0$ oraz $MD[h, e] = 1$.
- Gdy $CF[h, e] = 1$, oznacza to, że h jest prawdziwe na pewno, bo $MB[h, e] = 1$ oraz $MD[h, e] = 0$.

- W pozostałych przypadkach wartość współczynnika CF należy do przedziału $(-1 \ldots 1)$.

Rysunek 4.10 na stronie 79 pokazuje sieć zależności współczynników pewności. Taka sieć może służyć do szybkiego i efektywnego przeprowadzenia procesu wnioskowania.

W celu jego przeprowadzenia, należy zdefiniować sposób propagacji współczynników pewności CF.

W przypadku, gdy mamy do czynienia z regułami o wielu przesłankach należy wpierw wyliczyć sumaryczną wartość CF dla takiej reguły:

- Jeżeli reguła jest postaci `Jeżeli` e_1 `ORAZ` e_2 `to h ze stopniem pewności CF`, to sumaryczną wartość CF dla reguły wyliczamy w następujący sposób:

$$CF[h, e_1 \& e_2] = \min\{CF(e_1); CF(e_2)\} \cdot CF(h)$$

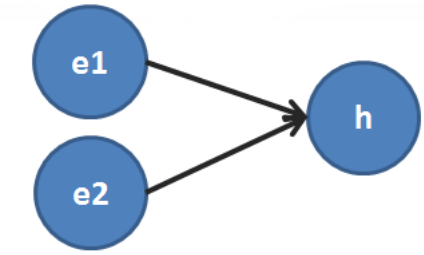
  Zapis ten oznacza wybór minimalnej wartości współczynnika CF spośród wszystkich przesłanek wchodzących w skład reguły, a następnie wyliczenie iloczynu z faktyczną wartością współczynnika CF konkluzji. W przypadku, gdy w regule znajduje się kilka konkluzji, brana pod uwagę jest więc ta najmniej słuszna.

- Jeżeli reguła jest postaci `Jeżeli` e_1 `LUB` e_2 `to h ze stopniem pewności CF`, to sumaryczną wartość CF dla reguły wyliczamy w następujący sposób:

$$CF[h, e_1 \& e_2] = \max\{CF(e_1); CF(e_2)\} \cdot CF(h)$$

  Tutaj brana pod uwagę jest maksymalna wartość CF ze względu na fakt, że taką regułę można rozbić na dwie oddzielne reguły, a w procesie wnioskowania wzięta pod uwagę byłaby właśnie reguła o wyższej wartości tegoż.

- W przypadku, gdy wszystkie przesłanki reguły uznajemy za pewne, niezależnie od ich liczby i sposobu połączeń logicznych pomiędzy nimi, do dalszego obliczania wykorzystywana jest wartość współczynnika CF konkluzji. Z taką sytuacją mamy do czynienia w przypadku większości systemów bazujących na tym podejściu.



Rysunek 4.11: Równoległe łączenie reguł

Rysunek 4.12: Szeregowe łączenie reguł

W przypadku, gdy dana hipoteza jest konkluzją więcej niż jednej reguły (patrz rysunek 4.11), należy zastosować jeden z poniższych wzorów na propagację niepewności wiedzy w zależności od wartości współczynników CF tych reguł (funkcja sgn to funkcja signum zwracająca liczbę -1 dla każdej liczby rzeczywistej ujemnej, 1 dla liczby rzeczywistej dodatniej, a 0 dla wartości 0).

Dla połączenia równoległego (rys. 4.11) należy zastosować jeden z poniższych wzorów:

$$sgn(CF[e_1,h]) = sgn(CF[e_2,h]) \Rightarrow$$
$$CF[e_1\ e_2,h] = CF[e_1,h] + CF[e_2,h] + |CF[e_1,h]| \cdot CF[e_2,h] \quad (4.4)$$

$$sgn(CF[e_1,h]) \neq sgn(CF[e_2,h]) \Rightarrow$$
$$CF[e_1\ e_2,h] = \frac{CF[e_1,h] + CF[e_2,h]}{1 - \min\{|CF[e_1,h]|;|CF[e_2,h]|\}} \quad (4.5)$$

Szeregowe łączenie reguł (rys. 4.12) wymaga jedynie wyliczenia iloczynu wartości współczynników CF:

$$CF[e_1\ e_2,h] = CF[e_1,h] \cdot CF[e_2,h]$$

Przedstawiony tutaj sposób propagacji współczynników CF nie jest jedynym aktualnie rozpatrywanym. Szczegółowe informacje, wraz z porównaniem oraz badaniami efektywności znaleźć można w pracy [91].

Dla podanego wcześniej przykładu, zakładając, że wszystkie przesłanki są prawdziwe, sumaryczna wartość współczynnika CF wynosi:

$$CF(S\ I, M) = 0,5 + 0,5 - 0,5 \cdot 0,5 = 0,75$$
$$CF(S\ I\ W, M) = 0,75 + 0,5 - 0,5 \cdot 0,75 = 0,875$$
$$CF(S\ I\ W\ M, P) = 0,875 \cdot 0,9 = 0,7875$$
$$CF(S\ I\ W\ M\ H, P) = \frac{0,7875 - 0,3}{1 - 0,3} = 0,696$$

Do niewątpliwych zalet metody współczynników pewności należy prosty system ich propagacji. Dzięki temu, faktyczna implementacja systemu korzystającego z tego podejścia jest prosta. Model ten w sposób racjonalny wydaje się przekuwać niepewność rozumianą przez człowieka na postać zinformatyzowaną. Po dostrojeniu i rozbudowaniu, systemu Mycin osiągnął dokładność klasyfikacji przewyższającą lekarzy uznawanych za ekspertów dziedzinowych. Stosunkowo prosty wydaje się być też proces akwizycji wiedzy.

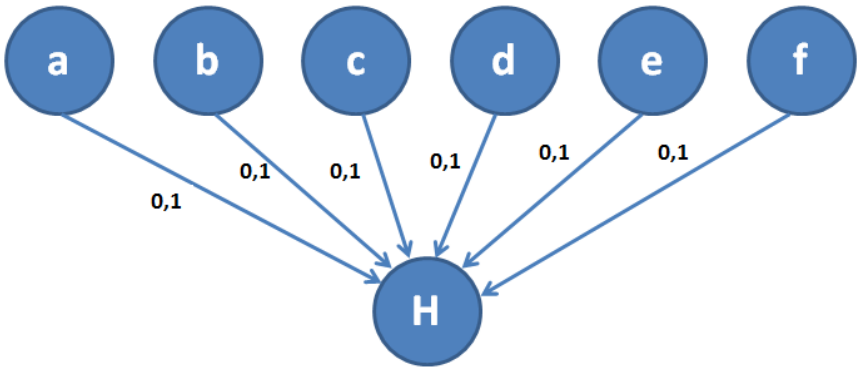
Niestety, model ten nie został w sposób formalny udowodniony i nie posiadał poprawnej podbudowy teoretycznej.

Kolejnym podejściem była meteda Dempstera-Shafera, która pozwoliła na pewne połączenie modeli probabilistycznych i podejściem opartym na modeli współczynników pewności CF.

W przypadku aktualizacji modelu, czasami konieczna jest aktualizacja wartości wszystkich współczynników CF, nawet tych już poprawnie wyliczonych. Największe jednak wady obejmują sam proces propagacji współczynników pewności. Prześledźmy przykład podany na rysunku 4.13. Widać tutaj sześć przesłanek (a do f) i jedną konkluzję H. Stopnie pewności tychże przesłanek są bardzo niskie i wynoszą w każdym przypadku 0,1. Porównać to można do sytuacji w której hipoteza H oznacza uzyskanie miliona dolarów, a poszczególne przesłanki zdarzenia małoprawdopodobne: wygrana na loterii, znalezienie pieniędzy na ulicy, darowizna, itp.

Jednakże ze względu na fakt, że przesłanek jest sporo - całkowita wyliczona wartość współczynnika CF hipotezy H wynosi 0,47. Jeśli przesłanek tych byłoby więcej, wartość ta będzie odpowiednio wyższa.

Drugim przykładem obrazującym słabość tego podejścia jest spadek wartości współczynnika CF dla długich łańcuchów wnioskowania (rys. 4.14). Jak widać w przykładzie, pomimo stosunkowo wysokich wartości współczynników CF na poszczególnych etapach, całkowita wartość CF hipotezy wynosi nawet mniej niż w poprzednim przykładzie ($CF[e_1 - e_5, h_1] = 0,43$).



Rysunek 4.13: Wiele przesłanek o małej wartości współczynnika CF

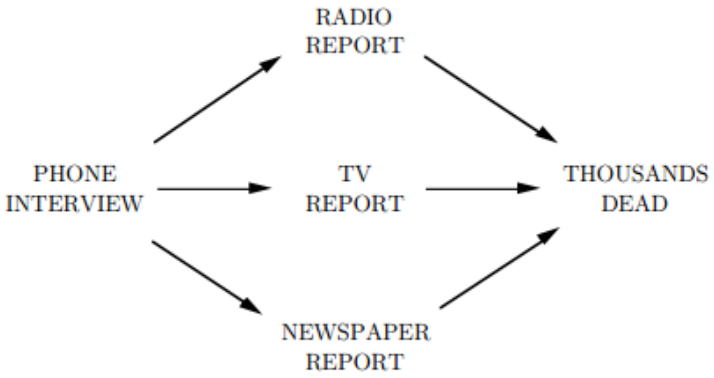
Rysunek 4.14: Długi łańcuch wnioskowania

Reguły logiki klasycznej zachowują zasadę modus ponens oraz zasadę lokalności (ang. *Principle of locality*). Oznacza to, że mając regułę logiczną "Jeżeli e to h" i wiedząc, że e jest prawdziwe - na mocy zasady modus ponens można powiedzieć, że h również jest prawdziwe; zasada lokalności mówi, że h będzie prawdziwe niezależnie od innych informacji, które jeszcze posiadamy.

Niestety, wnioskowanie w warunkach niepewności często narusza te dwie reguły. Stąd użycie modelu CF często prowadzi do błędów we wnioskowaniu [62, 121].

Kolejny problem przedstawiony jest na schematycznym rysunku pokazującym wnioskowanie w sprawie awarii elektrowni w Czarnobylu (4.15, źródło: [60]). Widać tutaj, że wszystkie doniesienia radiowe, telewizyjne oraz prasowe miały źródło w jednym wywiadzie telefonicznym. Tak przedstawiony model współczynników CF znacznie zwiększa nasze przekonanie o tym, że tysiące osób zmarło. Jeśli jednak otrzymamy informację, że wszystkie te komunikaty miały dokładnie jedno źródło – nasze przekonanie o słuszności tej hipotezy maleje.

Te oraz inne wady spowodowały, iż autor postanowił rozwinąć tę metodę



Rysunek 4.15: Schemat wnioskowania po awarii elektrowni w Czarnobylu

i zaproponować autorską metodę współczynników niepełności (ang. *incompleteness factor*) omówioną w dalszej części pracy.

4.8 Podsumowanie

Należy pamiętać, że nie każda wiedza da się w sposób dokładny zamodelować w wybranych metodach reprezentacji wiedzy. Niestety, żadna z nich nie jest doskonała do reprezentacji wiedzy niepełnej. Teoria Bayesa wymaga definiowania bardzo dużej liczby współczynników oraz zakłada niezależność zmiennych losowych. Logika rozmyta to konieczność definiowywania funkcji przynależności dla każdej zmiennej lingwistycznej i kosztowny matematycznie sposób wyliczania konkluzji. Zbiory przybliżone jedynie w sposób bardzo ogólny dają informacje o ostatecznej konkluzji. Teoria Dempstera-Shafera oraz współczynniki pewności CF wydają się być metodą obiecującą, jednakże również nie pozbawioną wad. W tym celu autor rozprawy proponuje użycie współczynników niepełności IF (ang. *incompleteness factor*) bazujących właśnie na współczynnikach CF. Jak to zostanie wykazane w kolejnych rozdziałach, metoda ta wydaje się niwelować przedstawione wady (w postaci relatywnie zbyt niskiej wartości współczynnika CF dla konkluzji przy propagacji niepewności wzdłuż długich ścieżek wnioskowania oraz relatywnie wysokich przy użyciem wielu źródeł wiedzy). Koncepcja współczynników niepełności wiedzy (IF) proponowana w rozprawie w połączeniu z ideą grupowania reguł podobnych do siebie w skupienia i przeglądania potem (w procesie wnioskowania) tylko reprezentantów skupień pozwalają na

optymalizację SWD pod względem szybkości wnioskowania jak i ilości oraz jakości wyprowadzonej nowej wiedzy.

Rozdział 5

Hierarchiczna struktura bazy wiedzy

Gwałtowny rozwój narzędzi i mechanizmów informatycznych spowodował ogromny przyrost liczby informacji, jakie są gromadzone i przetwarzane w systemach komputerowych. Większość z nich jednak jest zbierana w sposób automatyczny i w postaci nieprzetworzonej jest dla człowieka bezpośrednio niedostępna. Koronnym przykładem tutaj mogą być zarówno automatyczne logi serwerowe, zbierane z zachowaniem milisekundowych okienek czasowych, jak również informacje o zachowaniach klientów wielkich sieci handlowych. Informacje od danych różni jedna ważna cecha: informacja jest uporządkowana, jest zestawem użytecznych cech wydobytych z danych poprzez ich analizę i grupowanie, jest swego rodzaju ich podsumowaniem [153]. Dopiero od połowy ubiegłego wieku zaczyna się stosować narzędzia informatyczne w celu zamiany danych na informacje. Do tej pory proces ten wykonywany był całkowicie ręcznie, co w pewnym momencie historii przestało być ekonomicznie uzasadnione. Rozwinęła się więc dziedzina informatyki zwana analizą (ekstrakcją) danych (ang. *data mining*), która przenosiła ciężar tego żmudnego procesu w stronę komputerów.

Data mining (eksploracja danych) to proces ekstrakcji, odkrywania wcześniej nieznanych, niewidocznych na pierwszy rzut oka wzorców, potencjalnych powiązań i innych użytecznych informacji w dużych zbiorach danych [168]. Proces powinien być automatyczny lub częściej półautomatyczny, a odkryte wzorce powinny być prawdziwe, możliwe do zweryfikowania i miarodajne, wnosić nową wartość do informacji generując pewien zysk, zazwyczaj ekonomiczny. Mianem data miningu określa się również proces

predykcji, czyli generowania informacji, które bezpośrednio nie występują w analizowanych danych. Proces ten pozwala przewidzieć nowe wartości danych lub np. odtworzyć dane uszkodzone, zamazane na podstawie analizy tych już zgromadzonych i odkrytych w nich wzorców.

Eksploracja danych jest pojęciem bardzo szerokim [114]. W jej skład wchodzą m.in. następujące elementy:

Streszczanie (ang. *data generalization*) ma na celu stworzenie krótkiego, zwięzłego podsumowania streszczanych danych. Opis ten jest skoncentrowany na wcześniej wybranych pożądanych cechach. Przykładem tutaj może być streszczenie i podsumowanie wyników badań laboratoryjnych na dużej grupie ludzi.

Wykrywanie asocjacji (ang. *association rule learning*) to proces odkrywania powiązań pomiędzy danymi niewidocznych bezpośrednio. Najczęściej spotyka się ten typ eksploracji danych w dużych sieciach handlowych, które na podstawie wzorców behawioralnych klientów odkrywają towary najczęściej kupowane razem (powiązane ze sobą).

Wykrywanie odchyleń (ang. *outlier detection*) to odnajdywanie danych, które z jakichś powodów nie pasują do pozostałego zbioru. Wykorzystanie obejmuje nie tylko detekcję fałszerstw bankowych i ubezpieczeniowych, ale także wykrywanie nieprawidłowości w sekwencjonowanym łańcuchu DNA.

Analiza przebiegów czasowych (ang. *time series detection*), gdzie główną rolę odkrywają cykliczne podobieństwa zachowania się obserwowanej próbki. Analiza giełdowa wykorzystuje tą część data miningu.

Klasyfikacja wzorcowa (ang. *supervised classification*) czyli przypisywanie danych do wcześniej znanych klas lub kategorii. Najczęściej wykorzystywane przez sektor ubezpieczeniowy, gdzie potencjalnego klienta przypisuje się do wcześniej opracowanych grup ryzyka.

Klasyfikacja bezwzorcowa (ang. *unsupervised classification*) będąca procesem podobnym do powyższego, lecz tutaj nie ma określonych z góry kategorii, a zadaniem algorytmu analizy danych jest stworzenie nowej taksonomii i przypisanie analizowanych przypadków do wygenerowanych grup. Inaczej nazywamy ją taksonomią.

Grupowanie (ang. *clustering*) będące odmianą klasyfikacji bezwzorcowej, gdzie dane przypisywane są do grup na podstawie cech wspólnych, zwykle na podstawie wybranej wcześniej miary podobieństwa. Liczba grup, jak również opis ich reprezentanta nie są znane *a priori* i muszą być odkryte w trakcie działania algorytmu.

Eksploracja tekstów i WWW (ang. *text-mining*) obejmuje techniki wyszukiwania klasyfikacji i grupowania dokumentów tekstowych. Rozpatrywać tu będziemy wszelkie teksty, które nie są specjalnie utworzone do celów analizy automatycznej (np. logi systemowe, reguły w systemach wspomagania decyzji, teksty na stronach www, itd.).

Przebieg procesu odkrywania wiedzy ma charakter iteracyjny. Nieodzownym pierwszym elementem jest zrozumienie zagadnienia przez analityka danych, co pozwala również na łatwiejszą komunikację z ekspertem dziedzinowym, a co za tym idzie – na skuteczniejszą analizę danych. Drugi z etapów, to podział zbioru danych na dwa lub trzy zbiory (treningowy, testujący oraz ewentualnie walidacyjny). W ten sposób można budować rozwiązania w oparciu o pierwszy ze zbiorów danych, podczas gdy faktyczne testowanie i ewentualna walidacja przeprowadzona zostanie na innych danych, lecz ciągle z tej samej dziedziny i tego samego źródła. Po ograniczeniu zestawu danych, następuje proces wstępnego czyszczenia i przetwarzania danych. Ma tutaj miejsce zarówno konwersja danych do wybranej metody reprezentacji wiedzy, jak również wstępne usuwanie wartości izolowanych lub uznanych za nieprawidłowe [81]. Dokonywana jest również dyskretyzacja zmiennych lub też ich normalizacja czy też konwersja do jednego, przyjętego formatu (np. jedna skala mierzenia temperatury). Kolejnym stadium jest ewentualne ograniczenie liczby wymiarów stawianego problemu. Analizowanie każdej informacji o obiektach znacząco spowalnia cały proces, dlatego tak ważne jest wsparcie eksperta w wyborze atrybutów kluczowych dla danego problemu. Istnieją również metody służące do automatycznego lub półautomatycznego wyboru cech kluczowych, np. metoda składowych głównych [65]. W następnym etapie wybierane jest konkretne zadanie eksploracji danych spośród omówionych wcześniej zastosowań. Mając dane te informacje, można przystąpić do wyboru i implementacji konkretnego algorytmu i systemu pozwalającego na osiągnięcie zakładanych celów.

Nieodzownym elementem podsumowującym jest próba interpretacji znalezionych wzorców i weryfikacja ich wraz z ekspertem domenowym. Powoduje to, iż zebrana wiedza będzie nietrywialna i użyteczna z punktu widzenia

użytkownika końcowego. Ostatni etap obejmuje zebranie wniosków i przygotowanie ich w wersji czytelnej dla użytkownika końcowego.

5.1 Grupowanie danych

Analiza skupień jest jedną z podstawowych technik używanych w data miningu [98]. Metoda ta zmienia uporządkowanie obiektów w taki sposób, że obiekty podobne do siebie (zgodnie ze wcześniej zdefiniowaną miarą podobieństwa lub odległości) trafiają do tych samych grup (skupień), a obiekty różniące się od siebie – do różnych grup. Wspomniana funkcja podobieństwa jest więc kluczowym elementem technik grupowania, albowiem od jej jakości wydatnie zależy jakość otrzymanych grup.

Od lat '60 ubiegłego wieku techniki analizy skupień znajdują coraz szersze zastosowanie, a algorytmy stosowane w tym celu ulegają ciągłej ewolucji. W ostatnich czasach występuje wyraźna tendencja do specjalizowania narzędzi do konkretnego problemu zamiast prób uzyskania rozwiązania ogólnego, możliwego do zastosowania w szerokim spektrum przypadków. Autor tej rozprawy postanowił jednak zaproponować rozwiązanie na dużym poziomie ogólności, cechujące się możliwościami optymalizacyjnymi pod kątem szczegółowych zastosowań.

Jak napisano wcześniej, w procesie eksploracji danych ogromnie istotna jest weryfikacja poprawności wyników. Ma to co najmniej tak samo duże znaczenie w przypadku wykorzystywania algorytmów analizy skupień. Grupowanie danych nie może być traktowane jako proces w pełni automatyczny, lecz zwykle półautomatyczny. Kluczowe jest przetestowanie wielu podejść, dostrojenie parametrów oraz właściwa interpretacja i ewaluacja wyników. Osobnym problemem jest forma danych wejściowych, która np. dla danych przestrzennych musi zostać dostosowana do potrzeb konkretnego algorytmu grupującego. Szerzej problem analizy danych przestrzennych został omówiony w pracy [68].

Oprócz samego algorytmu podstawowymi parametrami koniecznymi do zdefiniowania są funkcja odległości, zakładana liczba skupień lub kryterium stopu algorytmu. Jak będzie to wykazane później, istnieją automatyczne lub półautomatyczne metody do wyznaczania niektórych z nich. Przykładowo, autor opracował i przystosował metodę wyznaczania optymalnej liczby skupień dla grupowania reguł algorytmem hierarchicznym AHC [107].

Ze względu na tak duże zróżnicowanie dostępnych algorytmów również

kryterium ich podziału nie jest tylko jedno. Najczęściej stosowanym kryterium podziału jest podział na algorytmy [16]:

- Hierarchiczne
  - Aglomeracyjne
  - Deglomeracyjne (podziałowe)
- Niehierarchiczne
  - K-means i podobne
  - K-medoids i podobne.

Inne kryterium podziału opiera się na strukturach danych wykorzystywanych w czasie procesu grupowania [75]:

Modele połączeniowe wykorzystujące hierarchiczne struktury danych, np. drzewa. Modele te oparte są na funkcji odległości (lub podobieństwa).

Modele oparte na centroidach, czyli obiektach uznawanych za reprezentatywne dla danej grupy. Przykładem są algorytmy k-means.

Modele oparte na rozkładzie wykorzystujące matematyczne funkcje rozkładu statystycznego. Algorytm oczekiwania-maksymalizacji [31] jest tutaj przykładem.

Modele gęstościowe takie jak DBSCAN [34] oraz OPTICS [7] wykorzystujące funkcję gęstości w celu obliczania odległości pomiędzy obiektami tworzącymi skupienia.

Modele grafowe korzystające z algorytmów grafowych (m.in. do znajdowania kliki w grafie).

Pozostałe niedające się zakwalifikować do żadnej z powyższych grup.

Kolejnym kryterium jest podział na algorytmy twarde, w których obiekt zawsze jest przynależny do jednej i tylko jednej grupy (skupienia) oraz algorytmy rozmyte, w których obiekt należy do wielu grup, a odpowiedni współczynnik określa stopień przynależności [92].

Ostatnim rozpatrywanym kryterium jest podział na algorytmy deterministyczne, w których wynik grupowania jest zawsze taki sam dla tego

samego zestawu danych oraz algorytmy stochastyczne (niedeterministyczne), w których czynnik losowy decyduje o ostatecznym podziale na grupy. Przedstawicielem tej pierwszej grupy jest szczegółowo omówiony algorytm AHC, natomiast drugiej – algorytm k-means.

Autor stosował różne algorytmy analizy skupień [67, 68], a doświadczenia z tych badań zaowocowały ostatecznie użyciem algorytmów deterministycznych, dostarczających stabilnych i stałych podziałów obiektów na grupy. Istnieją również inne badania rozpatrujące ten problem [106], jednakże w żadnym z dotychczasowych opracowań nie był brany pod uwagę wpływ niepełności wiedzy na skuteczność i efektywność wnioskowania. W tym celu autor tej rozprawy, po wielu badaniach i strojeniu parametrów początkowych, przystosował algorytm hierarchiczny AHC do grupowania reguł w SWD. Algorytm ten korzystając z mechanizmów analizy skupień daje również pewną informację o sposobie postępowania z niepełnością wiedzy. Problem ten będzie szczegółowo omówiony w rozdziale 5.3.3 na stronie 119.

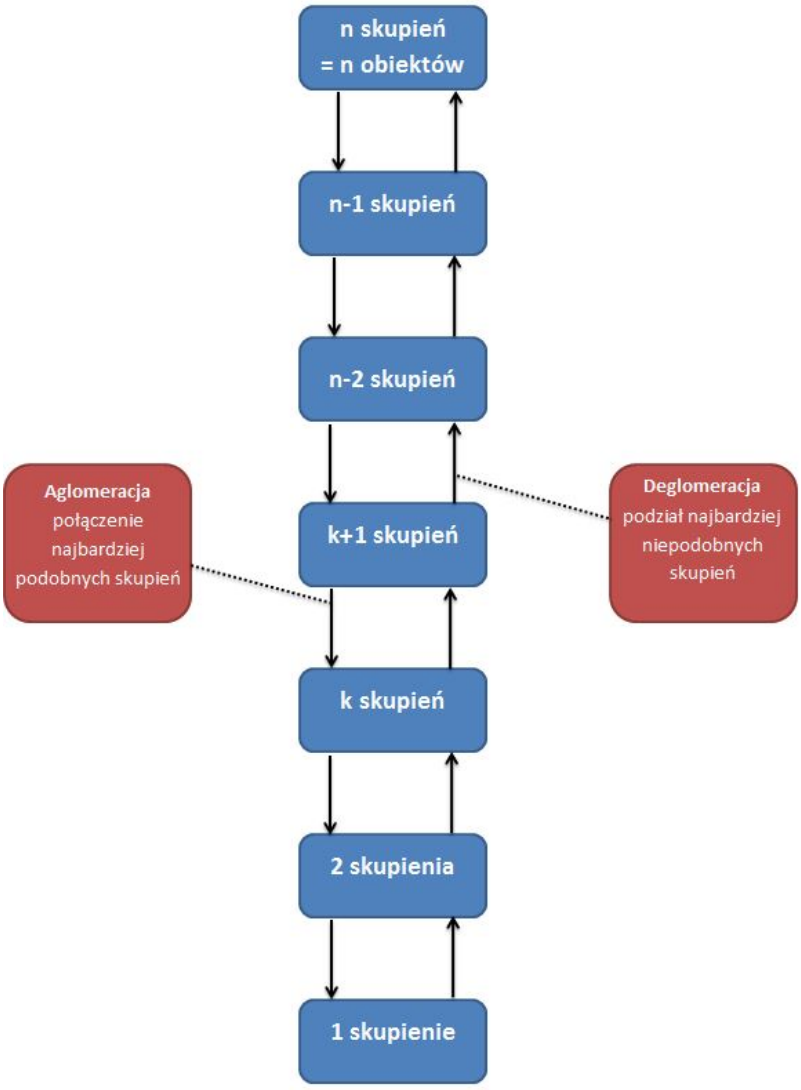
5.1.1 Algorytmy hierarchiczne

Algorytmy hierarchiczne charakteryzują się tym, że budują grupy obiektów krok po kroku. Poza tym, budują, omawianą dokładniej w dalszej części pracy, hierarchię skupień. W oryginalnej wersji dla n obiektów tworzą $n-1$ skupień. W obrębie tej grupy algorytmów wyróżnia się dwie podgrupy [76]:

- Algorytmy aglomeracyjne,
- algorytmy podziałowe (deglomeracyjne).

Algorytmy podziałowe rozpoczynają pracę z założeniem, że wszystkie obiekty w bazie stanowią jedną grupę. Następnie, w poszczególnych krokach, dzielą obiekty na poszczególne zagnieżdżone grupy, wewnątrz których podobieństwa obiektów są większe niż na poziomie wyższym. Algorytmy te kończą swój przebieg, gdy na najniższym poziomie otrzymamy tyle grup ile jest obiektów (innymi słowy: każdy obiekt będzie w oddzielnej grupie).

Algorytmy aglomeracyjne mają odwrotny przebieg: na początku zakłada się, że każdy z obiektów jest reprezentantem oddzielnej grupy. Następnie, w kolejnych krokach, algorytm próbuje odnaleźć obiekty, które są najbardziej do siebie podobne (lub najmniej do siebie niepodobne). Po ich odnalezieniu, zostaje utworzona nowa grupa obiektów składająca się z tych najbardziej podobnych (najmniej niepodobnych). Krok ten powtarza się aż do momentu, w którym wszystkie obiekty znajdują się w jednej grupie.



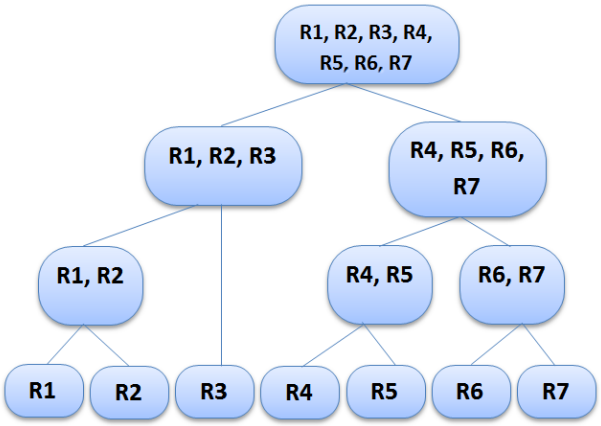
Rysunek 5.1: Aglomeracja i deglomeracja

Wadą algorytmów hierarchicznych jest duża złożoność obliczeniowa (rzędu $O(n^2)$, a nawet $O(n^3)$).

Wymagają one do poprawnego działania generowania macierzy podobieństwa (lub niepodobieństwa). Struktura ta to macierz trójkątna dolna, w której znajduje się tyle samo wierszy co kolumn. Na przekątnej macierzy znajdują się zwykle same zera lub wartość nieskończoności. Zarówno wiersze jak i kolumny są etykietowane identyfikatorami obiektów wchodzących w skład bazy danych podlegającej grupowaniu. Na przecięciu *i-tej* kolum-

ny oraz *j-tego* wiersza znajduje się wartość funkcji podobieństwa (niepodobieństwa) obiektu *i-tego* do *j-tego*. Im ta wartość większa (mniejsza) tym obiekty bardziej do siebie podobne.

Skuteczność algorytmów hierarchicznych zależy w ogromnym stopniu od wyboru sposobu wypełniania macierzy podobieństwa (czyli od wybranej metody wiązania obiektów w grupy). W wyniku działania algorytmów grupujących powstają zagnieżdżone struktury drzewiaste, tzw. dendrogramy (przykład przedstawia rys. 5.2 na str. 94). Są one naturalnym sposobem wizualizacji pogrupowanych danych.



Rysunek 5.2: Przykładowy dendrogram dla 7 reguł

Algorytmy aglomeracyjne

Pierwszym i najprostszym algorytmem aglomeracyjnym jest algorytm Agnes [76] nazywany również czasem AHC (ang. *Aglomerative Hierarchical Clustering*). Jego przebieg jest następujący:

1. Przypisz każdy obiekt do oddzielnej grupy. Tym samym, utwórz n grup, gdzie n to liczba obiektów grupowanych.
2. Utwórz grupę z dwóch grup o najmniejszej wielkości miary niepodobieństwa (największej mierze podobieństwa) z poprzedniego kroku.
3. Oblicz miarę podobieństwa (niepodobieństwa) pomiędzy nowoutworzoną grupą a wszystkimi pozostałymi grupami.
4. Powtarzaj kroki 2 i 3 aż do utworzenia jednej grupy zawierającej wszystkie obiekty.

Do obliczenia miary podobieństwa pomiędzy grupami stosuje się miary przedstawione w tabeli 5.7 na stronie 114.

Pewnym wariantem algorytmu jest przerwanie procesu grupowania w odpowiednim momencie. Autorka prac [106, 113] proponuje zatrzymanie procesu aglomeracji (łączenia) skupień w przypadku, gdy podobieństwo między skupieniami staje się większe niż między elementami wewnątrz danego skupienia. Oznaczałoby to bowiem, że różnice między grupami przestają być wyraźnie i trudne będzie ich efektywne przeszukiwanie. Okaże się przykładowo, że dwa różne skupienia mają już bardzo podobnych reprezentantów i wybór jednego z nich jako bardziej odpowiedniego będzie albo bardzo trudny albo niemożliwy. Z tego względu autorka bazując na kryterium podobieństwa wewnątrz skupień oraz niepodobieństwa między skupieniami omówionym w pracy [155] przez Theodoridisa i Koutroumbas określiła optymalny moment na zakończenie procesu grupowania. Wynikiem proponowanego w pracach [106, 113] algorytmu mAHC będzie zatem pewna grupa skupień o strukturze hierarchicznej wewnątrz każdego z nich (skupień) zamiast jednej dużej grupy, jak to ma miejsce w przypadku użycia klasycznego algorytmu AHC.

Autor niniejszej rozprawy poczynił dodatkowo starania do zaproponowania poprawnego sposobu wyznaczania kryterium stopu dla algorytmu *mAHC* użytego do grupowania reguł. Konkretne rozwiązanie wraz z pseudokodem prezentowane jest na stronie 123.

Przykład algorytmu *AHC* Prześledźmy bieg algorytmu na przykładzie. Mamy daną macierz niepodobieństwa daną w tabeli 5.1.

	1	2	3	4	5	6
1	-					
2	26	-				
3	26	32	-			
4	27	2	2	-		
5	6	26	41	45	-	
6	50	31	31	19	38	-

Tabela 5.1: Macierz niepodobieństwa dla przykładowych danych

Macierz ta jest wyliczana w pierwszym kroku za pomocą funkcji odległości (podobieństwa). Jak widać w przykładzie, nie zastosowano skalowania lub normalizacji danych. Nie jest to procesem koniecznym, jednakże jak

okaże się w trakcie badań – normalizacja ułatwia i przyspiesza proces odnajdywania reguł, a tym samym – zwiększa efektywność wnioskowania. Do przykładowego grupowania użyjemy metody średniego wiązania.

W pierwszym kroku zakładamy, że $n = 6$, jako że mamy 6 obiektów, które chcemy pogrupować.

W następnym kroku przeszukujemy tabelę w poszukiwaniu najmniejszych wartości. Minimalna wartość to 2. Występuje dla obiektów:

$$d(2,4) = d(3,4) = 2$$

W takim przypadku grupujemy pierwszy w kolejności napotkany obiekt. Powstaje grupa złożona z obiektów 2 oraz 4 (tabela 5.2 na stronie 96 obrazuje sposób powstania tabeli w kolejnym kroku. Tabela wynikowa to: 5.3 na stronie 96).

	1	2,4	3	5	6
1	-				
2,4	(26+27)/2	-			
3	26	(32+2)/2	-		
5	6	(26+45)/2	41	-	
6	50	(31+19)/2	31	38	-

Tabela 5.2: Tworzenie macierzy niepodobieństwa w drugim kroku algorytmu

	1	2,4	3	5	6
1	-				
2,4	26,5	-			
3	26	17	-		
5	6	35,5	41	-	
6	50	25	31	38	-

Tabela 5.3: Macierz niepodobieństwa po trzecim kroku algorytmu

Macierz niepodobieństw zmniejszyła swój stopień o jeden, powstała nowa grupa złożona z obiektów $2, 4$. Następnie należy powtarzać kroki 2 oraz 3 aż wszystkie obiekty zostaną zgrupowane.

Po zakończeniu działania algorytmu widać w jaki sposób zostały tworzone grupy. W wyniku otrzymaliśmy jedną grupę zawierającą wszystkie obiekty poddawane grupowaniu.

	1,5	2,4	3	6
1,5	-			
2,4	31	-		
3	33,5	17	-	
6	44	25	31	-

Tabela 5.4: Macierz niepodobieństwa po zgrupowaniu obiektów 1,5

	1,5	2,4,3	6
1,5	-		
2,4,3	31,42	-	
6	44	27	-

Tabela 5.5: Macierz niepodobieństwa po zgrupowaniu obiektów 2,4,3

	1,5	2,4,3,6
1,5	-	
2,4,3,6	34,57	-

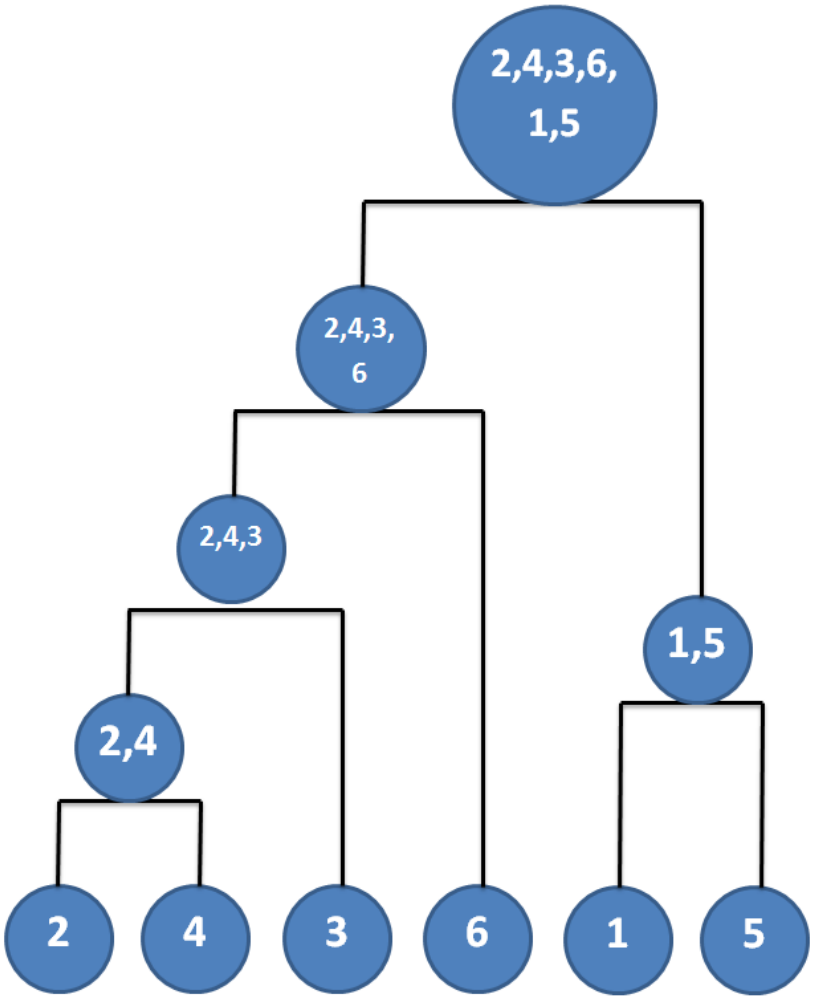
Tabela 5.6: Macierz niepodobieństwa po zgrupowaniu obiektów 2,4,3,6

Efektem działania algorytmu *AHC* dla tego zbioru przykładowego będzie dendrogram przedstawiony na rysunku 5.3. Przycinając odpowiednio drzewo (lub przerywając proces grupowania w odpowiednim momencie) możemy otrzymać grupy `(1,5); (2,4,3); (6)` lub `(1,5); (2,4,3,6)` w zależności od wysokości cięcia.

BIRCH Bardzo ciekawym algorytmem, którego nie można jednoznacznie zaklasyfikować do jednej z podanych grup, jest algorytm BIRCH [78]*. Produkuje on hierarchiczne drzewo obiektów, jednak w trakcie jego działania wykorzystuje również techniki niehierarchiczne. Algorytm sprawdza się przy przetwarzaniu dużych ilości danych ponieważ w trakcie swojego działania reaguje na możliwość przepełnienia pamięci operacyjnej wpisywanymi danymi.

Idea tego algorytmu jest podobna do koncepcji fraktoryzacji. Tutaj również obiekty z bazy danych są przyporządkowywane do podgrup, zwanych "cechami klastrów" (ang. *"cluster-features"*) . Te podgrupy są następnie grupowane w K grup za pomocą tradycyjnych, hierarchicznych metod grupowania.

*Balanced Interative Reducing using Cluster Hierarchies



Rysunek 5.3: Dendrogram dla przykładowych danych

"Cecha klastrów" (w skrócie CF) to w istocie zbiór informacji statystycznych o podzbiorze danych:

$$CF = \{M, LS, SS\} = \{M, \sum_{i=1}^{M} X_i, \sum_{i=1}^{M} X_i^2\}$$

gdzie M oznacza liczbę elementów w danym skupieniu, LS sumę cech obiektów, a SS sumę kwadratów cech obiektów.

Powyższe informacje są wystarczające do poprawnego przeprowadzenia procesu grupowania. W jego wyniku powstaje struktura zwana CF-drzewem. Obiekty są grupowane iteracyjnie, jeden obiekt po drugim

umieszczany jest w drzewie (można zauważyć pierwszą wadę algorytmu - wrażliwość na kolejność danych wejściowych). O rozmiarach i strukturze CF-drzewa decydują dwa parametry: B oraz ϵ.

Drzewo CF składa się z węzłów będących liściami lub nie będących liściami. Te ostatnie mają co najwyżej B potomków, którzy reprezentują cechy klastrów CF_i dla $i = 1, \ldots, B$. Węzeł nie będący liściem reprezentuje grupę złożoną z podgrup jego potomków. Węzeł będący liściem zawiera co najwyżej L obiektów, z których każdy obiekt jest cechą klastra CF_j dla $j = 1, \ldots, L$. Węzły-liście reprezentują grupy utworzone ze wszystkich członków grupy.

Parametr ϵ określa maksymalną średnicę grupy. Jeśli po dodaniu nowego obiektu, średnica nowej grupy przekroczy tą wartość, następuje rozbicie grupy. Wybierane są dwa najbardziej oddalone od siebie obiekty, a następnie wszystkie obiekty z grupy poprzedniej otrzymują nowe przyporządkowanie do jednej z dwóch grup.

Sam algorytm BIRCH [92] przedstawia się następująco:

1. Zanalizuj dane i zbuduj CF-drzewo.
2. Jeśli w czasie konstrukcji drzewa przekroczona zostanie maksymalna wartość pamięci operacyjnej, to wtedy potraktuj liście obecnego drzewa jak obiekty. Zwiększ parametr ϵ i zbuduj następne drzewa (tym razem mniejsze) z "cech klastrów" i pozostałych, nieprzetworzonych jeszcze obiektów.
3. Powtarzaj kroki 1 i 2 aż do zbudowania pełnego CF-drzewa, które mieści się w pamięci.
4. Za pomocą metody hierarchicznej dokonaj grupowania CFów w k grup.
5. Dokonaj przypisania obiektów do najbliższych im skupień.

Algorytmy deglomeracyjne

Przeciwieństwem algorytmu Agnes jest algorytm Diana [76]. Zamiast budować od podstaw hierarchię grup, rozbija on wstępną, złożoną ze wszystkich obiektów grupę tak, aby każdy obiekt był w końcu oddzielną grupą.

Algorytm składa się z $n - 1$ kroków, gdzie n to liczba obiektów. W każdym kroku algorytmu następuje wybór grupy C, która jest wewnętrznie najmniej spójna (zawiera obiekty najbardziej niepodobne do siebie). Parametr ten, określany mianem średnicy, jest definiowany jako:

$$diam(C) := \max_{i,j \in C} d(i,j),$$

gdzie $d(i,j)$ to odległość pomiędzy dwoma obiektami i oraz j z grupy C. Sam podział przebiega w sposób następujący:

1. Do zbioru A przypisz wszystkie elementy grupy C. Za zbiór B przypisz zbiór pusty.

2. Przenieś jeden obiekt ze zbioru A do B w następujący sposób:

   a) Dla każdego obiektu $i \in A$ oblicz współczynnik $\alpha(i)$, średnią miarę niepodobieństwa to wszystkich innych obiektów grupy A.

   b) Wybierz obiekt m o największej wartości współczynnika α i przenieś go do grupy B. Wtedy:

   $$A := A \backslash \{m\},\ B := \{m\}$$

3. Przesuń inne obiekty z A do B:

   a) Jeśli $|A| = 1$, wtedy koniec.

   b) W przeciwnym przypadku oblicz dla wszystkich $i \in A, \alpha(i)$.

   c) Oblicz średnie niepodobieństwo obiektu i do wszystkich obiektów z B (oznaczane przez $d(i,B)$).

   d) Wybierz obiekt $h \in A$ dla którego:

   $$\alpha(h) - d(h,B) = \max_{i \in A} (\alpha(i) - d(i,B))$$

   e) Jeśli $\alpha(h) - d(h,B) > 0$ wtedy przesuń obiekt h z A do B i wróć do punktu 3.

   f) W przeciwnym wypadku, zatrzymaj proces i pozostaw zbiory A oraz B w nienaruszonym stanie.

5.1.2 Algorytmy niehierarchiczne

W przeciwieństwie do algorytmów hierarchicznych, algorytmy niehierarchiczne (podziałowe, partycjonujące) wymagają na samym początku podania docelowej liczby skupień. Liczba ta jest kluczowa i stała podczas działania algorytmu. Co więcej, żadna z utworzonych grup nie może pozostać pusta po wykonaniu procesu grupowania.

Zasadą działania algorytmów partycjonujących jest wstępne podzielenie zestawu danych na K grup, a następnie iteracyjna poprawa tego podziału aż do momentu stwierdzenia braku możliwości poprawy.

Jak można łatwo wyliczyć, sprawdzenie wszystkich możliwości podziału n obiektów na K grup wyrażone liczbą Stirlinga drugiego rzędu [42]:

$$S_n^{(K)} = \frac{1}{K!}\sum_{i=0}^{K}(-1)^{K-i}\binom{K}{i}i^n$$

jest ekstremalnie trudne już dla małych wartości K oraz n (przykładowo, dla $K = 5$ oraz $n = 25$ liczba możliwości to 2436684974110751).

Istnieje wiele metod i algorytmów niehierarchicznych, wśród których do najważniejszych należą:

1. K-means - każda grupa jest reprezentowana przez jej centroid.
2. K-medoids - każda grupa jest reprezentowana przez jeden z jej elementów (medoid).

Algorytmy używające metod niehierarchicznych są algorytmami niedeterministycznymi, nieoptymalizowane mają również tendencję do wpadania do minimów lokalnych funkcji grupowania. Ich zaletami jest jednak dużo mniejsza złożoność obliczeniowa (rzędu $O(n)$). Niestety, k-means są bardzo podatne na szumy i wartości znacznie odbiegające od średnich.

Algorytm k-means

Algorytm k-means jest klasycznym przedstawicielem algorytmów niehierarchicznych [92]. Jego działanie polega na losowym podziale wejściowego zbioru obiektów X na k wzajemnie rozłącznych, niepustych podzbiorów (grup). Parametr k musi być zdefiniowany jeszcze przed działaniem algorytmu, co stanowi jego znaczącą wadę (ponieważ nierzadko analityk danych nie wie ile naturalnych skupień występuje w analizowanym zbiorze).

Algorytm początkowo w sposób stochastyczny wybiera tzw. centroidy, czyli obiekty stanowiące "środki" grup. Środek grupy jest tutaj rozumiany jako punkt w przestrzeni m wymiarowej (m to liczba atrybutów opisujących obiekty) o najmniejszej odległości w stosunku do pozostałych elementów wchodzących w skład grupy. Odległość (podobieństwo) rozumiana jest w taki sam sposób jak w przypadku algorytmów hierarchicznych.

Po wylosowaniu początkowych centroidów, algorytm w sposób iteracyjny stara się wyznaczyć nowe środki grup. Jeśli mu się to uda, obiekty są

ponownie przypisywane do grup tak, aby znaleźć się najbliżej środków grup. Klasycznym kryterium stopu jest sytuacja, w której nie da się poprawić już dopasowania obiektów lub też osiągnięta zostanie zadana wcześniej liczba iteracji algorytmu.

Algorytm k-means charakteryzuje się niską złożonością obliczeniową na poziomie $O(n)$, jednakże jest algorytmem niedeterministycznym, a co za tym idzie – wyniki otrzymywane są trudne do powtórzenia. Często algorytm uruchamia się wielokrotnie aby zminimalizować wpływ losowości na wyniki grupowania, co niestety zwiększa jego czas działania.

Zapis algorytmu prezentuje się następująco:

Algorytm 3: Algorytm k-means

Dane: $X = \{X_1 \dots X_n\}$ – obiekty grupowane; K – założona liczba skupień

Rezultat: $C = \{C_1 \dots C_K\}$ - K grup obiektów wraz z $c_1 \dots c_K$ centroidami

begin

  $c_1 \dots c_K := X_{random}$ przyjmij losowe obiekty jako centroidy;

  Przypisz pozostałe obiekty X_i do najbliższych skupień $C_1 \dots C_K$;

  while *nastąpiły zmiany w skupieniach* **do**

    $c_1 \dots c_K :=$ oblicz nowe centroidy na podstawie odległości obiektów wchodzących w skład danej grupy;

    Przydziel obiekty $X_1 \dots X_n$ do najbliższych skupień $C_1 \dots C_K$

  end

end

Algorytm k-medoids

Drugim algorytmem niehierarchicznym jest k-medoids [76]. Zasada działania jest bardzo podobna do algorytmu k-means, jednakże w tym przypadku zamiast centroidów występują tzw. medoidy. Medoid będzie definiowany jako element z początkowego zbioru obiektów wybierany jako reprezentant grupy. W odróżnieniu od k-means, gdzie środkiem grupy mógł być punkt spoza oryginalnego zbioru danych, w algorytmie k-medoids występuje ograniczenie do wyboru reprezentantów skupienia jako obiektów z oryginalnego zbioru poddawanego grupowaniu.

Najbardziej popularną odmianą algorytmu k-medoids jest PAM (ang. *Partitioning Around Medoids*). Początkowe działania algorytmu są tożsame z k-means. W sposób losowy wybieranych jest k obiektów uznawanych za medoidy, a pozostałe obiekty przypisywane są do grupy zgodnie z naj-

bliższym medoidem. Następnie każdy z wybranych medoidów zamieniany jest z obiektami wchodzącymi w skład aktualnego skupienia. Obliczany jest koszt takiej zamiany i spośród wszystkich wariantów – wybierany jest taki o najmniejszej całkowitej wartości funkcji kosztu definiowanej jako suma odległości obiektów wewnątrz grupy do obiektu wybranego jako medoid. Dzięki temu procesowi wybierany jest obiekt leżący najbliżej w stosunku do pozostałych elementów skupienia.

Po wybraniu medoidów proces dopasowywania pozostałych obiektów do grup jest powtarzany. Algorytm kończy swoje działanie w momencie w którym w dwóch kolejnych iteracjach nie wystąpiły żadne zmiany w podziale na grupy lub też wyczerpano limit iteracji algorytmu.

Pseudokod algorytmu ma następującą postać:

Algorytm 4: Algorytm k-medoids

Dane: $X = \{X_1 \dots X_n\}$ – obiekty grupowane; K – założona liczba skupień

Rezultat: $C = \{C_1 \dots C_K\}$ - K grup obiektów wraz z $m_1 \dots m_K$ medoidami

```
begin
    m_1 ... m_K := X_random przyjmij losowe obiekty jako medoidy;
    Przypisz pozostałe obiekty X_i do najbliższych skupień C_1 ... C_K;
    while nastąpiły zmiany w skupieniach do
        foreach medoid m_i do
            foreach X_j niebędącego medoidem do
                Zamień obiekty m_i oraz X_j ze sobą i oblicz koszt
                zamiany
            end
        end
        Wybierz dopasowanie o najmniejszym koszcie;
    end
end
```

5.2 Parametry grupowania

Algorytmy analizy skupień stworzone były pierwotnie do przetwarzania danych o dużym rozmiarze. Tak postawione zadanie implikuje pewne pożądane cechy idealnego algorytmu grupującego [93]:

- Minimalizacja liczby iteracji algorytmu, wręcz tworzenie grup w czasie jednego przebiegu działania.

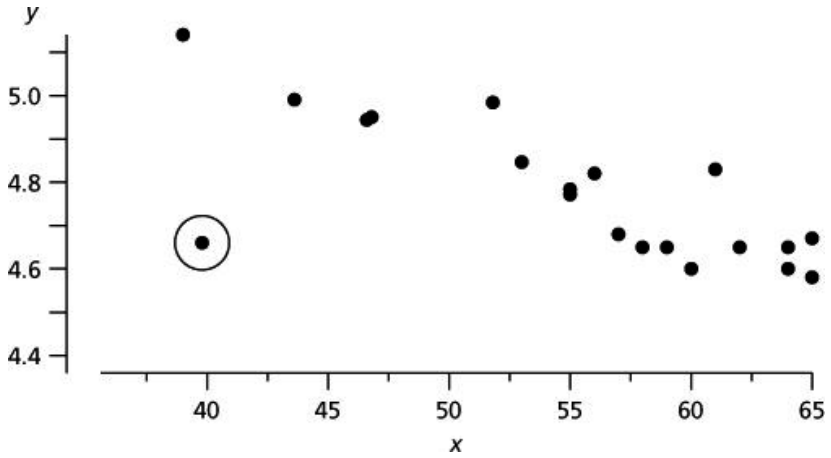
- Niewrażliwość na kształt skupień danych, a co za tym idzie - zdolność do tworzenia grup o dowolnie skomplikowanych kształtach (w tym zagnieżdżonych).
- Niewrażliwość na szum informacyjny w danych i wartości izolowane.
- Samouczenie się, możliwość pracy nienadzorowanej. Parametry takie jak liczba grup nie powinny być wymagane; sam algorytm powinien wykryć najbardziej właściwą liczbę grup na podstawie analizowanych danych.
- Umiejętność grupowania różnych typów danych: binarnych, dyskretnych i kategorycznych (zbiory danych mogą zawierać informacje różnego typu, a algorytm powinien umożliwiać grupowanie każdego z nich).
- Niewrażliwość na kolejność przetwarzania informacji (dla każdej możliwej permutacji obiektów, wyniki powinny być identyczne).
- Skalowalność (algorytm powinien poprawnie działać dla danych wielowymiarowych, zarówno dla małych i dużych zbiorów danych).
- Mała złożoność obliczeniowa (techniki grupowania są z założenia stosowane do dużych i bardzo dużych zbiorów danych, często opisanych przez wiele parametrów, więc nawet drobne obniżenie złożoności obliczeniowej finalnie może poskutkować realnym skróceniem czasu obliczeń).

Niestety do chwili obecnej nie został opracowany algorytm, który spełniałby wszystkie te warunki dla każdego rodzaju danych i należy doświadczalnie dobierać odpowiedni algorytm do analizowanych informacji do momentu osiągnięcia zadowalających rezultatów.

5.2.1 Wartości izolowane, szum informacyjny

W czasie przeprowadzania procesu grupowania nie wszystkie wartości dają się jednoznacznie zakwalifikować do utworzonych grup. Obiekty, nierzadko opisane licznym zbiorem atrybutów, mogą mieć w swoim opisie wartości spoza zakresu uznawanego za typowy. Sytuacja ta zwana szumem informacyjnym wpływa na jakość tworzonych skupień. Szum informacyjny powstaje za sprawą błędów pomiarowych, błędów konwersji danych, czy też błędów ludzkich. W dużych bazach danych jest nie do wychwycenia za pomocą ręcznych metod analizy. Czasami zdarza się również, że na którymś etapie

wartość danego atrybutu zostaje pominięta z różnych powodów. W tej sytuacji mamy do czynienia z problemem wiedzy niepełnej lub też z brakiem w danych (ang. *missing values*).



Rysunek 5.4: Graficzna interpretacja wartości odstającej

Sytuacja zobrazowana na rysunku 5.4 przedstawia sytuację występowania tzw. wartości izolowanych (ang. *outlier*) [3]. Obiekty te w sposób skuteczny mogą doprowadzić do zaburzenia procesu grupowania, zwłaszcza w algorytmach, które nie są odporne na występowanie takich wartości (np. k-means). Ze względu na wykorzystanie funkcji podobieństwa (odległości) tego typu odstające dane mogą zaburzyć proces tworzenia się grupy. Przykładowo, w sytuacji gdyby w algorytmie k-means za centrum grupy przyjąć taką wartość izolowaną, odległość pomiędzy środkiem grupy a dowolnym innym obiektem byłaby zbyt duża aby poprawnie sformować grupę. Lepiej z tym problemem radzą sobie algorytmy hierarchiczne, w których to taka odstająca wartość stanowić będzie oddzielną gałąź na dendrogramie i nie zaburzy procesu grupowania pozostałych obiektów.

Istnieje kilka metod służących do poprawnego modelowania systemu z wartościami izolowanymi:

Retencja wartości odstających, czyli ich wykrycie i oznaczenie do przejrzenia przez eksperta. W ten sposób to człowiek ostatecznie podejmuje decyzję, czy wartość obserwacji jest na tyle istotna (i możliwa), że należy ją brać pod uwagę w procesie grupowania, czy też jest błędem i należy ją usunąć.

Wykluczenie (usunięcie) odchyleń w sposób automatyczny jest kontrowersyjnym sposobem na uniknięcie błędów dopasowania elementów

do grup. Należy zauważyć, że jest to ryzykowne i może doprowadzić do utracenia informacji istotnych z punktu widzenia procesu ekstrakcji danych. Automatyczne usuwanie wartości izolowanych ma miejsce zwykle tylko w systemach o bardzo dużej złożoności, gdzie manualna weryfikacja jest niemożliwa. Za każdym razem jednak należy poinformować użytkownika końcowego o przyjętej metodologii usuwania wartości odstających.

Zmiana podejścia jest konieczna dla zbiorów danych, których rozkład nie pokrywa się z rozkładem normalnym. W grupowaniu nigdy nie można założyć, iż grupy będą miały jakiś z góry upatrzony kształt (np. kulisty). W tym celu należy przeprowadzić eksperymenty i w przypadku otrzymania niesatysfakcjonujących danych – zmienić używany algorytm grupujący.

Zastosowanie podejścia rozmytego w którym dany obiekt może należeć do kilku różnych klas (grup) z inną wartością przynależności. W ten sposób niejako problem jest omijany i przekazywany człowiekowi do interpretacji. Należy jednak pamiętać, że podobnie jak w systemach ekspertowych – propagowanie niepewności wiedzy w postaci rozmywania przynależności obiektu do grupy (np. łączenie skupień ze sobą) zwielokrotnia błąd.

W ostatnich latach pojawiło się pojęcie "eksploracji wartości odstających" (ang. *outlier mining*) [6]. Jest to gałąź data miningu, która zajmuje się analizą wartości izolowanych. Zgodnie z nowymi odkryciami, analiza taka powinna poprzedzać sam proces ekstrakcji danych, ponieważ sama w sobie jest w stanie dostarczyć nowych i użytecznych informacji.

5.2.2 Rola reprezentanta

Po przeprowadzeniu procesu grupowania analityk danych zawsze staje przed problemem interpretacji i wizualizacji danych. Dodając do tego fakt, iż analiza skupień jest zwykle wykonywana dla baz zawierających bardzo dużo obiektów – należy się zastanowić w jaki sposób dokonać analizy każdej otrzymanej grupy.

Z rozważaniami tymi wiąże się problem reprezentanta skupienia. Pojęciem tym będziemy nazywać wektor cech najlepiej charakteryzujących obiekty wchodzące w skład danego skupienia. Zamiennie, używa się również określenia centroid (inaczej obiekt centralny) [9].

Poprawny wybór stosowanego reprezentanta jest ogromnie istotny z punktu widzenia późniejszej analizy. Najczęstszym celem jest tutaj wspomniana wcześniej charakterystyka danej grupy. Jednakże wyznaczenie centroidów może również wspomóc proces klasyfikacji kolejnych obserwacji do istniejących już skupień. Wystarczy w tym celu porównać dodawany obiekt do wyznaczonych centroidów i zdecydować w której grupie umieścić nowy obiekt. Posiadanie poprawnie wyznaczonych reprezentantów pozwoli nam na istotne skrócenie czasu przeglądu bazy wiedzy. Oprócz tego, mając danych reprezentantów skupień na każdym poziomie grupowania w hierarchicznym algorytmie aglomeracyjnym, możemy skorzystać z algorytmu wyboru węzła najbardziej obiecującego opisanego przez Saltona [135]. Sposób jego wykorzystania w kontekście wnioskowania wykorzystującego skupienia reguł zostanie opisany w dalszej części pracy.

W zależności od rodzaju grupowanych danych, centroid może przybierać różne formy. Zwykle dla danych typu numerycznego stosowana jest wartość średnia z wartości opisujących dane. Niestety, wiadomym jest, iż średnia nie zawsze dobrze przybliża i opisuje elementy ją tworzące. Przykładowo, dla wartości skupionych przy minimalnej wartości z danego zbioru oraz jednej, ekstremalnie wysokiej wartości danej zmiennej, wyliczona średnia będzie nienaturalnie przesunięta w kierunku tej dużej wartości właśnie. W przypadku, gdy średnia (arytmetyczna, ważona, harmoniczna, itd.) nie będzie dobrym sposobem na opis elementów danej grupy, stosuje się inne metody. Dla wartości numerycznych często stosuje się inne metody statystyczne opisu obiektu. Alternatywą dla średniej może być mediana (czyli wartość środkowa niemalejąco posortowanego zestawu danych) lub też dominanta (czyli wartość numeryczna najczęściej występująca w zbiorze danych).

Inaczej kwestię reprezentanta należy rozwiązać dla cech mających wartości kategoryczne (jakościowe). Wektor centroidalny powinien wtedy jak najbliżej odzwierciedlać wartości występujące w grupie, którego będzie reprezentantem. Najczęściej stosowaną formą reprezentanta jest wektor cech występujących najczęściej (moda) lub też wektor cech wspólnych dla wszystkich elementów wchodzących w skład grupy. W niektórych zastosowaniach stosuje się również wektor zawierający wszystkie cechy obiektów wchodzących w skład grupy. Takie podejście jednak powoduje znaczne wydłużenie opisu reprezentanta oraz generuje konflikty (ponieważ w jednej grupie mogą się zdarzyć obiekty o różnych, wzajemnie wykluczających się wartościach cech).

Do problemu grupowania reguł w systemach ekspertowych należy podejść jeszcze w inny sposób. Ze względu na tak szczególny charakter two-

rzenia reprezentantów, należy wykorzystać kombinację różnych sposobów. Autor w części eksperymentalnej dokonał analizy różnych metod tworzenia reprezentantów skupień, także pod kątem wykorzystania ich do wspomagania procesu wnioskowania w warunkach wiedzy niepełnej. Szczegółowe wyniki przedstawione są w dalszej części rozprawy.

Poniżej przedstawiona zostanie krótka charakterystyka metod tworzenia reprezentanta w kontekście grupowania reguł.

Centroid jako zbiór cech charakteryzujących wszystkie obiekty danej grupy

Centroid taki to inaczej przecięcie zbioru wartości atrybutów poszczególnych reguł (reprezentant tworzony za pomocą spójnika logicznego AND). Metoda ta wybiera wszystkie te cechy, które opisują każdą regułę wchodzącą w skład danego skupienia. Reprezentant ten cechuje się dużym zwarciem i krótką budową. Niestety, jak to zostanie wykazane w części eksperymentalnej, stosunkowo często zdarza się, że tworzone grupy zawierają dużą liczbę obiektów, które są wzajemnie do siebie bardzo podobne, lecz jako całość – nie przejawiają wielu identycznych cech. Ze względu na tą właściwość, reprezentant tworzony jako zbiór cech charakteryzujących każdy obiekt w grupie okazuje się być nieodpowiedni do zakładanego celu.

Centroid jako zbiór cech dominujących w opisach danej grupy

W przypadku, gdy grupy tworzone przez algorytm analizy skupień są duże, jednym ze sposobów na budowę reprezentanta jest wybór cech najczęściej występujących spośród wszystkich cech opisujących elementy danej grupy. Metoda ta z jednej strony – zachowuje zalety krótkiego opisu danej grupy (brak zwiększonej liczby cech w stosunku do elementu wchodzącego w skład grupy), z drugiej jednak dla grup heterogenicznych – może powodować zafałszowane wyniki. Należy jednak pamiętać, że głównym celem wykorzystania algorytmów analizy skupień jest generowanie spójnych grup, a co za tym idzie – reprezentant tworzony jako zbiór cech dominujących może poprawiać rezultaty grupowania.

W przypadku skorzystania z tej metody, należy się jednak zastanowić, czy cechy domiujące będą wyznaczane globalnie (tj. spośród wszystkich cech występujących we wszystkich regułach w systemie), czy w obrębie zbiorów wartości każdej cechy oddzielnie. Pierwsze podejście może wygenerować reprezentanta o kilku wartościach tego samego atrybutu, co znacząco utrudni późniejszą analizę takiej grupy.

Występowanie wiedzy niepełnej utrudnia wykorzystanie cech dominujących jako wektora centroidalnego. Może się zdarzyć sytuacja, w której jakaś kluczowa cecha ma sporo brakujących wartości. W tym przypadku wyznaczony reprezentant będzie nieprawidłowy.

Centroid jako zbiór cech unikalnych dla obiektów danej grupy

Sposobem na łatwe i duże rozróżnienie grup między sobą jest wybór cech charakteryzujących wyłącznie obiekty danej grupy. W ten sposób dokonywana jest dalsza separacja grup pomiędzy sobą, albowiem każda z nich jest opisywana tylko przez wartości unikalne. Niestety, sposób ten bardzo często gubi kluczowe informacje o wartościach cech, dzięki którym obiekty stworzyły skupienie. W przypadku grupowania reguł, reprezentant tego typu ma tendencję do wyboru cech niszowych, występujących ogromnie rzadko. Fakt ten znacząco utrudnia późniejsze wnioskowanie z użyciem utworzonych grup.

Centroid jako zbiór wartości atrybutów opisujących wszystkie obiekty wchodzące w skład danej grupy

Alternatywnym sposobem budowy reprezentanta jest użycie wszystkich pojęć występujących w opisie elementów wchodzących w skład danej grupy (innymi słowy - połączenie ich spójnikiem logicznym OR). Dzięki temu centroid opisuje dokładnie wszystkie reguły wchodzące w skład grupy. Pewną niedogodnością jest znaczna długość takiego reprezentanta, lecz z punktu widzenia późniejszej analizy – grupy reprezentowane przez taki centroid są łatwiejsze do analizy i przeprowadzania późniejszego wnioskowania dając relatywnie lepsze wyniki.

5.2.3 Miary odległości i podobieństwa

Wynik działania algorytmów analizy skupień bardzo mocno zależy od poprawnie przyjętej miary podobieństwa obiektów do siebie. Jej wartość może być określana za pomocą różnych metryk, a także za pomocą miary odległości pomiędzy dwoma obiektami. Miara podobieństwa może przyjmować różny charakter [42]:

- miary odległości,
- współczynnika podobieństwa,

- miary asocjacji.

Miary podobieństwa

Miara podobieństwa traktowana jest jako funkcja:

$$p : \Omega \times \Omega \rightarrow \Re^{+}$$

taka, że:

$$\begin{aligned} 0 \leqslant p(x,y) < 1 \text{ dla } x \neq y, \\ p(x,y) = 1 \text{ dla } x = y, \\ p(x,y) = p(y,x). \end{aligned}$$

W literaturze określa się ją także jako miara bliskości, zgodności [42]. Należy zauważyć, że maksymalna wartość podobieństwa równa 1 oznacza nierozróżnianie dwóch obiektów ze sobą i jest przeciwieństwem do miary odległości, gdzie znormalizowana odległość równa 1 oznacza całkowite przeciwieństwo dwóch obiektów. Przyjmuje się również, że:

$$p(x,y) = \frac{1}{d(x,y)}$$

gdzie $d(x,y)$ to odległość pomiędzy dwoma obiektami.

Dla problemu grupowania, pojęcia odległości i podobieństwa są często używane zamiennie, należy pamiętać tylko o ich przeciwnym znaczeniu (wysokie podobieństwo jest tożsame z małą odległością).

Wśród powszechnych miar podobieństwa należy wymienić:

Miarę Gowera, którą można stosować zarówno dla obiektów opisanych cechami ilościowymi oraz jakościowymi. Miara ta posiada zdefiniowany współczynnik wagowy w_{x_l,y_l} przyjmujący wartość 0, gdy wartość l-tej zmiennej nie jest znana dla jednego lub obu obiektów x i y oraz 1 w przeciwnym wypadku. We wzorze występuje również wartość funkcji podobieństwa l-tej zmiennej s_{x_l,y_l} określanej w zależności od typu zmiennej l. Liczba zmiennych określających obiekty to z. Funkcja ta może być dobierana dla każdej cechy osobno. Wzór na miarę Gowera przedstawia się następująco:

$$p_{gower}(x,y) = \frac{\sum_{l=1}^{z} s_{x_l,y_l} \cdot w_{x_l,y_l}}{\sum_{l=1}^{z} w_{x_l,y_l}}$$

Miarę kosinusową wyznaczającą stopień podobieństwa obiektów x oraz y na podstawie z atrybutów opisujących te obiekty daną wzorem:

$$p_{cos}(x,y) = \frac{\sum_{l=1}^{z} x_l y_l}{\sqrt{\sum_{k=1,l=1}^{z} x_l^2 \cdot \sum_{l=1}^{z} y_l^2}}$$

Miarę nakładania wyznaczającą stopień podobieństwa obiektów x oraz y na podstawie z atrybutów opisujących te obiekty daną wzorem:

$$p_{ovlap}(x,y) = \frac{\sum_{l=1}^{z} \min(x_l, y_l)}{\min(\sum_{l=1}^{z} x_l^2, \sum_{l=1}^{z} y_l^2)}$$

Miary odległości

Funkcja będąca miarą odległości między dwoma obiektami x oraz y ma postać [42]:

$$d : \Omega \times \Omega \to \Re^+$$

i spełnia warunki dla metryk:

$$\begin{aligned} d(x,y) = 0, \text{ gdy } x = y \\ d(x,y) \geqslant 0 \\ d(x,y) = d(y,x) \\ d(x,z) \leqslant d(x,y) + d(y,z). \end{aligned}$$

Jak widać, miara odległości nie ma określonej górnej granicy. Z tego powodu w praktycznych zastosowaniach stosuje się jednak skalowanie ich wartości do przedziału $[0 \ldots 1]$. Odwrotnie niż przy miarach podobieństwa, mniejsza odległość oznacza, że obiekty leżą bliżej siebie (są zbliżone swoim opisem).

Najczęściej stosowane miary odległości przedstawione są poniżej. We wszystkich poniższych wzorach wyliczana jest odległość d pomiędzy obiektami x oraz y na podstawie $1, 2, \ldots, i, \ldots, z$ cech (atrybutów) opisujących obiekty.

Odległość euklidesowa pomiędzy obiektami x oraz y na podstawie z atrybutów opisujących te obiekty wyznaczana jako:

$$d_{eukl}(x,y) = \sqrt{\sum_{i=1}^{z} (x_i - y_i)^2}$$

Odległość Manhattan pomiędzy obiektami x oraz y na podstawie z atrybutów opisujących te obiekty definiowana jest wzorem:

$$d_{manh}(x, y) = \sum_{i=1}^{z} |x_i - y_i|$$

Miara Manhattan nazywana jest również miarą miejską lub taksówkową. Jej nazwa wywodzi się z amerykańskiej dzielnicy Nowego Jorku, w której ulice przecinają się w równych odległościach pod kątem prostym. Miara ta jest najkrótszą odległością jaką trzeba pokonać aby dotrzeć z jednego punktu do drugiego poruszając się tylko wzdłuż poziomych i pionowych linii siatki.

Odległość Czebyszewa zwana jest inaczej miarą maksimum lub miarą szachową ze względu na to, że dla przestrzeni dwuwymiarowej jest to odpowiednik ilości ruchów, które musiałaby na szachownicy wykonać figura króla, by przemieścić się z jednego pola do drugiego:

$$d_{czeb}(x, y) = max_{i=1,\ldots,z}|x_i - y_i|.$$

Miara ta jako prosta obliczeniowo wydatnie skraca czas obliczeń.

Odległość Hamminga określa liczbę wartości atrybutów rozróżniających dwa obiekty. Przykładowo dla reguł:

```
1:JEŻELI Pogoda=Ładna ORAZ Pora roku=Lato TO Spacer=Tak
2:JEŻELI Pogoda=Ładna ORAZ Pora roku=Jesień TO Spacer=Nie
```

odległość Hamminga będzie wynosić 2[†].

W przypadku porównywania wartości binarnych, odległość Hamminga to liczność jedynek w wartości funkcji XOR wykonanej pomiędzy tymi wartościami.

Odległość Levenshteina (zwana także odległością edycyjną) jest numeryczną interpretacją liczby zmian koniecznych do przekształcenia jednego ciągu tekstowego w drugi. Metoda ta jest uogólnieniem miary Hamminga, lecz dopuszcza się tutaj ciągi o różnej długości (brakujące elementy są traktowane jak operacja usunięcia litery). Operacje traktowane jako przekształcenia (operacje proste) różnią się w zależności

[†]Przy założeniu, że wartość atrybutu decyzyjnego jest również brana pod uwagę przy rozróżnianiu obiektów. Szerzej o tym problemie można przeczytać w rozdziale 5.3

od konkretnej implementacji tej miary, ale zwykle są zdefiniowane jako: dodanie znaku, usunięcie znaku oraz zamiana znaku na inny. Czasami dopuszcza się również zamianę kolejności dwóch sąsiadujących znaków (lub słów w zdaniu) jako kolejną z operacji prostych. Przykładowo, odległość między słowami `klaster` a `klasa` wynosi 3, ponieważ należy zamienić literę "t" na "a" oraz usunąć litery "e" i "r".

Odległość Mahalanobisa Jeśli dane są dwa wektory wartości

$$[x] = \{x_1, x_2, \ldots, x_z\}[y] = \{y_1, y_2, \ldots, y_z\}$$

oraz pewna macierz symetryczna dodatnia określoną $G = cov(x_i, y_i)$ będącą macierzą kowariancji cech obiektów x oraz y, to wtedy odległość Mahalanobisa zdefiniowana jest jako:

$$d_m(\mathbf{x}, \mathbf{y}) := \sqrt{(\mathbf{x} - \mathbf{y})G^{-1}(\mathbf{x} - \mathbf{y})^T}.$$

Odległość ta może zostać wykorzystana w zagadnieniach grupowania danych, np. w grupowaniu rozmytym do określania kształtu grupy (skupienia) [55].

Odległość Minkowskiego będąca uogólnioną miara odległości między punktami przestrzeni euklidesowej. Odległość ta jest oznaczana L_m i definiowana wzorem:

$$L_m(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{z} |x_i - y_i|^m \right)^{1/m}.$$

Łatwo zauważyć, że dla różnych wartości parametru m odległość ta przyjmuje postać:

$m = 2$ odległości euklidesowej,

$m = 1$ odległości miejskiej,

$m \to \infty$ odległości Czebyszewa.

5.2.4 Kryteria łączenia skupień

Po zdefiniowaniu podobieństwa pomiędzy obiektami należy również zdefiniować kryteria łączenia skupień ze sobą.

$d^{new}_{(p+q),i}$ = Miara niepodobieństwa pomiędzy nową grupą $(p+q)$ a grupą i
d_{pi} = Miara niepodobieństwa pomiędzy grupami p oraz i
d_{qi} = Miara niepodobieństwa pomiędzy grupami q oraz i
$s^{new}_{(p+q),i}$ = Miara podobieństwa pomiędzy nową grupą $(p+q)$ a grupą i
s_{pi} = Miara podobieństwa pomiędzy grupami p oraz i
s_{qi} = Miara podobieństwa pomiędzy grupami q oraz i
$b_{q,p,i}$ = Współczynniki skalujące

Metoda	Wzór na wartość podobieństwa i niepodobieństwa
Całkowite wiązania	$d^{new}_{(p+q),i} = \max(d_{pi}, d_{qi})$; $s^{new}_{(p+q),i} = min(s_{p_i}, s_{q_i})$
Pojedyncze wiązania	$d^{new}_{(p+q),i} = \min(d_{p_i}, d_{q_i})$; $s^{new}_{(p+q),i} = \max(s_{p_i}, s_{q_i})$
Średnie wiązania	$d^{new}_{(p+q),i} = \frac{(d_{p_i}+d_{q_i})}{2}$
Średnie wiązania ważone	$d^{new}_{(p+q),i} = \frac{(b_p \cdot d_{p_i}+b_q \cdot d_{q_i})}{b_p+b_q}$
Wiązania za pomocą median	$d^{new}_{(p+q),i} = \frac{1}{2} \cdot d_{pi} + \frac{1}{2} \cdot d_{qi} - \frac{1}{4} \cdot d_{pq}$
Wiązanie za pomocą centroidów	$d^{new}_{(p+q),i} = \frac{b_p}{b_p+b_q} \cdot d_{pi} + \frac{b_q}{b_p+b_q} \cdot d_{qi} - \frac{b_p \cdot b_q}{(b_p+b_q)^2} \cdot d_{pq}$
Wiązanie Warda	$d^{new}_{(p+q),i} = \frac{1}{b_p+b_q+b_i} \cdot [(b_p+b_i) \cdot d_{pi} + (b_q+b_i) \cdot d_{qi} - b_i \cdot d_{pq}]$

Tabela 5.7: Kryteria łączenia skupień

W tabeli 5.7 przedstawione są najpopularniejsze kryteria służące do łączenia skupień [1, 145]. Przyjęto oznaczenia jak na stronie 114.

Użycie różnych metod wiązania powoduje zmianę zachowywania się algorytmu w stosunku do danych. I tak w przypadku metody całkowitego wiązania (ang. *complete linkage*) użyta jest funkcja max. Powoduje to, że tak zbudowane grupy są bardziej jednorodne, albowiem funkcja max bierze pod uwagę maksymalną odległość pomiędzy dwoma obiektami należącymi do grupy. Z tego też powodu, metoda ta jest również zwana "metodą najdalszego sąsiada".

Przeciwieństwem powyższego jest metoda pojedynczego wiązania (ang. *single linkage*). W tym przypadku używana jest funkcja min, która bierze pod uwagę odległość do najbliższego sąsiada. Metoda ta ma tendencję do

tworzenia małej liczby heterogenicznych grup, a w rezultacie - zmniejszenia efektywności wyszukiwania.

Metoda całkowitego wiązania może prowadzić do dylatacji - liczba grup może okazać się za duża. Jak to jednak będzie przedstawione w wynikach eksperymentalnych, duża liczba małolicznych grup będzie strukturą, w której wnioskowanie będzie łatwiejsze. Przypomnieć należy, że struktura drzewiasta pozwala na uniknięcie liniowego porównywania reprezentantów grup z aktualnym zbiorem faktów, a zamiast tego – korzysta z metody węzła najbardziej obiecującego.

Zarówno całkowite jak i pojedyncze wiązanie produkuje wartości, które są niewrażliwe na skalowanie, tzn. można je dla wygody pierwiastkować, logarytmować, a zależności i relacje pomiędzy nimi i tak zostaną zachowane.

Metoda średnich wiązań (ang. *Unweighted pair-group average*) odległość między dwoma skupieniami oblicza jako średnią odległość między wszystkimi parami obiektów należących do dwóch różnych skupień. Metoda ta jest efektywna, gdy obiekty formują naturalnie oddzielone "kępki", ale zdaje także egzamin w przypadku skupień wydłużonych, mających charakter łańcucha. Metoda średnich wiązań ważonych (ang. *Weighted pair-group average*) jest podobna do metody średnich połączeń, z tym wyjątkiem, że w obliczeniach uwzględnia się wielkość odpowiednich skupień (tzn. liczbę zawartych w nich obiektów) jako wagę. Zatem metoda ta (inaczej niż poprzednia) powinna być stosowana wtedy, gdy podejrzewamy, że liczebności skupień są wyraźnie nierówne. Metoda środków ciężkości (ang. *Unweighted pair-group centroid*) jest analogiczna do wyostrzania metodą środka ciężkości w logice rozmytej. Środek ciężkości skupienia jest średnim punktem w przestrzeni wielowymiarowej zdefiniowanej przez te wymiary. W metodzie tej, odległość między dwoma skupieniami jest określona jako różnica między środkami ciężkości. Metoda ważonych środków ciężkości (mediany) (ang. *Weighted pair-group centroid*) jest tożsamą do poprzedniej, z tym wyjątkiem, że w obliczeniach wprowadza się wagi, aby uwzględnić różnice między wielkościami skupień (tzn. liczbą zawartych w nich obiektów). Zatem, metoda ta jest lepsza od poprzedniej w sytuacji, gdy istnieją (lub podejrzewamy, że istnieją) znaczne różnice w rozmiarach skupień.

Metodą różniącą się od wszystkich pozostałych wydaje się być miara Warda. Do oszacowania odległości między skupieniami wykorzystuje podejście analizy wariancji. Zmierza się tu do minimalizacji sumy kwadratów dowolnych dwóch skupień, które mogą zostać uformowane na każdym etapie.

5.3 Grupowanie reguł w bazie wiedzy

Nadrzędnym celem tej rozprawy jest optymalizacja wnioskowania w systemach z wiedzą niepełną. Autor w tym celu postanowił wykorzystać mechanizmy analizy skupień w celu uprzedniego stworzenia hierarchicznej struktury bazy wiedzy, aby potem móc efektywnie wnioskować przy użyciu tak zbudowanej bazy wiedzy. Niniejszy podrozdział prezentuje w sposób szczegółowy proponowane rozwiązanie. Przedstawiono sposób przygotowania bazy wiedzy i jej hierarchiczną reprezentację. Następnie autor przedstawia sposób wnioskowania dla tak stworzonej struktury, aby przejść do omówienia autorskiej metody współczynników IF do modelowania niepełności wiedzy. Przedstawione zostają również rozważania nt. przystosowania metody węzła najbardziej obiecującego do działania w hierarchicznej strukturze bazy wiedzy zawierającej dane niepełne.

5.3.1 Przygotowanie danych

Klasyczne systemy wspomagania decyzji zawierają płaską strukturę bazy wiedzy, w której reguły zapisywane są w postaci klauzul Horna w dowolnej kolejności. Jak udowodniono wcześniej, taki zapis jest co najmniej nieefektywny i może powodować konieczność ciągłego przeszukiwania całej bazy wiedzy, co wykonywane będzie w czasie $O(n)$, gdzie n to liczba reguł.

System do poprawnego działania potrzebuje hierarchicznego zapisu wiedzy (innymi słowy: przekształca płaską strukturę bazy wiedzy w model hierarchiczny). Autor korzysta z reguł zapisanych w formacie zgodnym z pakietem RSES [13]. Każda z wygenerowanych reguł minimalnych ma postać:

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

Część konkluzyjna od przesłankowej oddzielona jest operatorem wynikania (=>), przesłanki połączone są spójnikiem logicznym "AND" oznaczanym jako "&". Baza wiedzy przygotowywana jest poprzez wygenerowanie reguł minimalnych, zgodnie z algorytmem przedstawionym w rozdziale 2.3.4 na stronie 26.

Należy zauważyć fakt, iż liczba deskryptorów wchodzących w skład reguł minimalnych będzie zawsze mniejsza lub równa liczbie atrybutów występujących w systemie. Utworzone reguły powinny być jak najbardziej ogólne i krótkie, a co za tym idzie – ich długość będzie zwykle mniejsza

od liczby atrybutów w systemie. W celu poprawnego zapisania niepewności autor proponuje pamiętanie wartości wszystkich atrybutów warunkowych. W przypadku wartości nieokreślonej lub nieustalonej proponuje się zastąpienie wartości atrybutu informacją o niezdefiniowaniu wartości `UND`. Dzięki temu przedstawiony sposób wnioskowania będzie mógł być zrealizowany w sposób bardziej efektywny. Dla podanego wyżej przykładu, reprezentacja taka miałaby postać:

```
UND & UND & UND & 8600 & UND & UND & UND & 177 & 2
151 & UND & UND & 8600 & UND & UND & UND & UND & 2
UND & UND & UND & 8600 & UND & UND & 30  & UND & 2
```

Ze względu na fakt oryginalnego występowania atrybutu decyzyjnego w różnych miejscach reguły, system ten traktuje atrybuty decyzyjne na tym etapie tak jak atrybuty warunkowe i nie rozgranicza informacji o nich. Różna liczba deskryptorów wchodzących w skład reguły nie będzie tutaj żadną przeszkodą, albowiem w przypadku nie występowania danej wartości atrybutu w konkretnej regule, zostanie zapamiętana wartość `UND`.

Autor rozpatruje atrybuty typu numerycznego jak i kategorycznego (symbolicznego) stąd każda informacja pamiętana w systemie będzie traktowana jako łańcuch znaków. Dzięki temu porponowane podejście będzie również odpowiednie dla baz wiedzy skonstruowanych z baz danych bez zamiany atrybutów symbolicznych na ich odpowiedniki numeryczne.

Tego typu reprezentacja wiedzy wydaje się prowadzić do większej zajętości pamięci, jednakże w opinii autora – zysk czasowy wprowadzony dzięki niej jest niewspółmiernie wyższy w porównaniu do strat przez nią generowanych. Warto podkreślić, iż w przypadku algorytmu RETE, omówionego wcześniej, autorzy również zdecydowali się na poświęcenie złożoności pamięciowej na rzecz optymalizacji czasowej. W tamtym przypadku jednak złożoność pamięciowa pozostaje na znacznie większym (kilka rzędów wielkości) poziomie niż w przypadku proponowanego rozwiązania. Dzięki zapisowi każdej reguły jako wektor $v_i = \{v_{a_1}, v_{a_2}, \ldots, v_{a_z}, v_{d_1}, \ldots, v_{d_p}\}$ złożony z z wartości atrybutów warunkowych i p wartości atrybutów decyzyjnych znacznie łatwiejsze będzie porównywanie reguł i późniejsze na nich wnioskowanie.

5.3.2 Struktura bazy wiedzy

W porównaniu do klasycznej struktury bazy wiedzy omówionej na początku tej rozprawy, autor proponuje zmodyfikowanie jej formalnego opisu zgodnie z formalnym opisem przedstawionym w rozdziale 2.4 na stronie 31.

Pierwowzorem do tworzonego rozwiązania był m.in. system SMART Saltona. Ideą jego działania było grupowanie dokumentów na podstawie ich podobieństwa między sobą [135]. Podobnie jak w proponowanym podejściu, wyszukiwanie odbywało się w znacznie krótszym czasie w stosunku do przeszukiwania liniowego, ze względu na wykorzystanie reprezentantów grup i struktury drzewiastej. W pracy proponuje się zastosowanie algorytmu grupującego AHC do stworzenia skupień złożonych z reguł tworzących bazę wiedzy. Poza znacznym przyspieszeniem wyszukiwania, uzyskuje się również informacje o grupach najbardziej podobnych reguł. Ta struktura danych pozwoli w dalszym etapie na odnalezienie grupy reguł najbardziej podobnych do aktualnego zbioru faktów. Dzięki temu, w sytuacji w której żadna z reguł nie może zostać uaktywniona, możliwym będzie uaktywnienie reguł najbardziej relewantnych przy jednoczesnym oznaczeniu wiedzy przez nie generowanej jako niepewnej.

Proponowane podejście różni się od modelu klasycznego SWD tym, że przechowuje reguły w strukturze drzewiastej ($Tree$) oraz określa podobieństwo tych reguł za pomocą funkcji f_{sim} zgodnie z jedną z metod omówionych wcześniej. Wybór najwłaściwszej metody został dokonany po przeprowadzeniu szeregu eksperymentów omówionych w dalszej części pracy.

Po odnalezieniu właściwego skupienia, nastąpi uaktywnienie reguł wchodzących w skład odnalezionego skupienia. Oczywistym jest, że nie wszystkie przesłanki w analizowanym skupieniu reguł będą miały pokrycie w zbiorze faktów. W kolejnym rozdziale przedstawiona jest propozycja oznaczenie wiedzy wygenerowanej przez takie reguły jako niepewnej poprzez dodanie do faktów dodatkowego współczynnika pewności. Fakty dopisywane do bazy wiedzy w wyniku uaktywnienia reguł dla których wszystkie przesłanki są prawdziwe (wiedza jest pełna) przyjmują stopień pewności z wartością optymalną równą 1. Natomiast, fakty dodane do bazy wiedzy w wyniku uaktywnienia reguły o niepełnej liczbie prawdziwych przesłanek (wiedza niepełna) będą przyjmować stopień pewności odpowiednio mniejszy. Każdy deskryptor wchodzący w skład zbioru faktów oznaczony będzie poprzez trójkę:

$$d_i =< a_i, v_{a_i}, CF(d_i) >$$
$$CF(d_i) \in [0 \dots 1], d_i \in F \text{ - zbiór faktów}$$

gdzie $CF(d_i)$ to współczynnik pewności i-tego deskryptora, $CF(f_j)$ to współczynnik pewności j-tego faktu będącego częścią wspólną zbioru deskryptorów D_i oraz zbioru faktów (F).

5.3.3 Grupowanie reguł

Grupowanie reguł nie będzie różniło się koncepcyjnie od grupowania innych danych zapisanych w postaci zbiorów deskryptorów. W przypadku grupowania reguł należy jednak przystosować nie tylko parametry algorytmu grupującego, ale również funkcję podobieństwa tak, aby radziła ona sobie z tymi specyficznymi danymi.

Jak napisano wcześniej, proponowane rozwiązanie będzie uniwersalne i możliwe do zastosowania do wartości atrybutów będących wartościami liczbowymi (numerycznymi) jak również kategorycznymi. Z racji faktu nie zakładania normalizacji[‡] bazy wiedzy przed przystąpieniem do grupowania, proponuje się użycie dwóch różnych funkcji podobieństwa (odległości):

Proste podobieństwo (SS) będące w istocie informacją o liczbie wspólnych deskryptorów wchodzących w skład dwóch reguł:

$$simpleSimilarity = card(D_p \cap D_q)$$

gdzie D_p oraz D_q to zbiory deskryptorów p-tej oraz q-tej reguły odpowiednio. Miara ta jak zostanie wykazane w eksperymentach nie sprawdza się zbyt dobrze ze względu na fakt faworyzowania reguł o dużej liczbie deskryptorów. Współczynnik SS jest nienormalizowany i może przyjmować wartości większe lub równe 0.

Ważone podobieństwo (WS) biorące pod uwagę również długość reguł, których podobieństwo jest liczone:

$$weightedSimilarity = \frac{card(D_p \cap D_q)}{card(D_p \cup D_q)}$$

[‡]Czyli zamiany danych symbolicznych na postać numeryczną.

Autor wybrał algorytm grupowania hierarchicznego AHC z kilku powodów:

1. AHC generuje hierarchię skupień, co pozwoli na szybkie wyszukiwanie reguł. Jeżeli za n przyjmiemy liczbę grupowanych reguł, to algorytm potrzebuje $2 \cdot n - 1$ kroków do wyznaczenia hierarchii skupień. Złożoność obliczeniowa procesu wyszukiwania w tej hierarchii jest na poziomie $O(log_2 n)$.

2. Algorytm AHC może wykorzystywać dowolne funkcje podobieństwa i kryteria łączeń skupień w grupy.

3. Wybrany algorytm jest odporny na wartości odstające i izolowane.

4. AHC jest algorytmem deterministycznym, wyniki działania nie zależą od kolejności przetwarzania reguł.

5. Algorytm w prosty sposób tworzy reprezentantów skupień na każdym etapie łączenia skupień w grupy. Fakt ten pozwoli na zaproponowanie kontr-podejścia korzystającego z jego zmodyfikowanej wersji mAHC.

Algorytm grupowania reguł

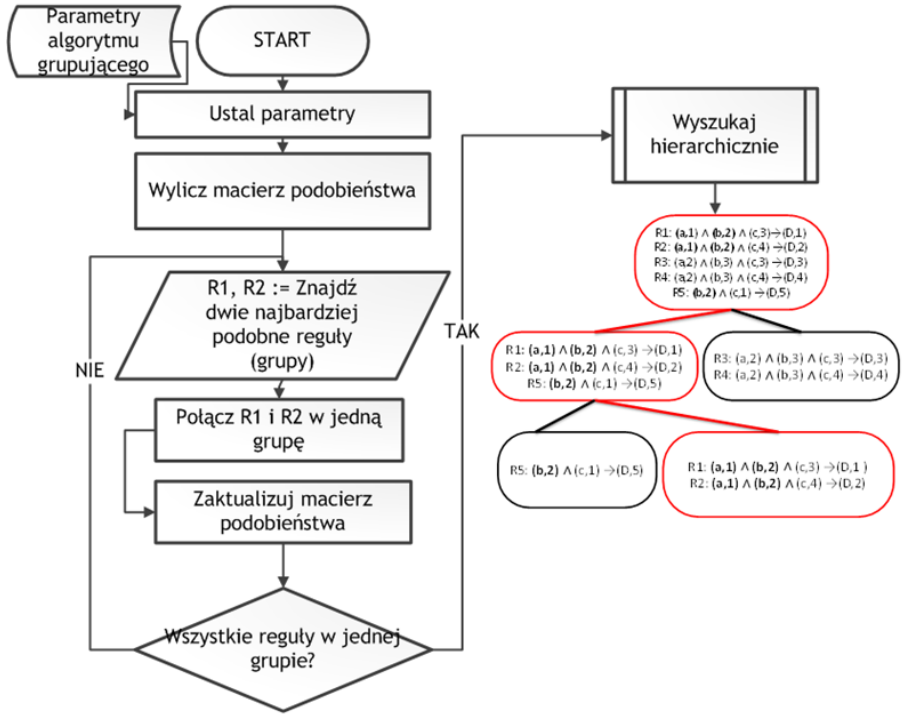
W pierwszym kroku algorytmu następuje generowanie kwadratowej macierzy podobieństwa, której rozmiar jest równy liczbie reguł zapisanych w systemie. Algorytm na przecięciu i-tego wiersza i j-tej kolumny wylicza podobieństwo dwóch reguł do siebie zgodnie ze sposobami omówionymi powyżej.

Po wyliczeniu macierzy podobieństwa, następuje właściwy algorytm grupowania. Wyszukiwana jest maksymalna wartość współczynnika podobieństwa określająca, które reguły połączyć. Następnym krokiem będzie wyliczenie wartości podobieństwa nowopowstałego skupienia do pozostałych reguł. Tutaj również algorytm pozwala na manipulacje sposobami jego wyliczania zgodnie z kryteriami łączenia skupień omówionymi wcześniej. W części eksperymentalnej autor bada wpływ tegoż kryterium na efektywność procesu wnioskowania.

Klasyczne podejście korzystające z algorytmu AHC buduje pełną hierarchię skupień i umożliwia wyszukiwanie reguł do uaktywnienia za pomocą metody węzła najbardziej obiecującego. Kontr-podejście korzystające z algorytmu mAHC implementuje autorski sposób wyznaczania kryterium stopu omówionego w części eksperymentalnej pracy.

W celu dalszego rozróżnienia reguł od siebie, autor traktuje klasę decyzyjną jako dodatkowy deskryptor, zarówno w przypadku liczenia podobieństwa dwóch reguł, jak również wyliczania reprezentanta grupy i wnioskowania. Podejście to jest uzasadnione faktem, iż wartość decyzji pamiętana w regule jest kluczowym elementem, nierzadko pozwalającym na poprawne podzielenie reguł w systemie.

Schemat blokowy algorytmu znajduje się na rysunku 5.5.



Rysunek 5.5: Schemat blokowy przedstawionego algorytmu grupującego reguły

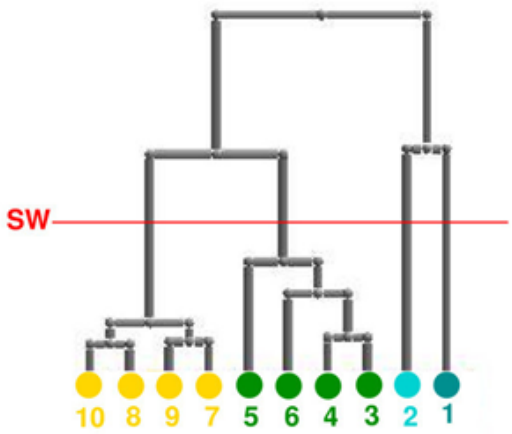
Algorytm w sposób formalny można zapisać następująco:

Algorytm 5: Algorytm AHC do grupowania reguł

Dane: $U = \{r_1 \dots r_n\}$ – obiekty (reguły) grupowane;
Rezultat: Dendrogram zawierający skupienia reguł
begin
  Ustal parametry algorytmu;
  Wyznacz macierz podobieństwa zgodnie z przyjętym kryterium odległości;
  while *Wszystkie reguły nie są w jednym skupieniu* **do**
    R_1, R_2 := Dwie najbardziej podobne reguły (skupienia) ;
    Połącz R_1 oraz R_2 w jedno skupienie zgodnie z przyjętym kryterium ;
    Zaktualizuj macierz podobieństwa;
  end
end

Algorytm mAHC oraz dobór kryterium stopu

Algorytm *mAHC* w wersji znanej z literatury przerywa proces grupowania na pewnym etapie. Dzięki temu zmniejsza się liczba kroków potrzebna do grupowania danych. W rozprawie prezentowana jest jednakże autorska modyfikacja tej koncepcji. Algorytm *mAHC* (algorytm nr 6) użyty w tej rozprawie dokonuje pełnego grupowania w celu uzyskania pełnego dendrogramu. W kolejnym kroku liczba skupień jest ograniczana poprzez przycięcie dendrogramu w sposób analogiczny do pokazanego schematycznie na rys. 5.6. Na przykładowym rysunku po dokonaniu przycięcia na poziomie *SW* otrzymujemy cztery grupy zaznaczone różnymi kolorami.



Rysunek 5.6: Proces przycinania dendrogramu

Algorytm 6: Algorytm mAHC wraz z automatycznym wyznaczeniem optymalnej liczby skupień

Dane: $U = \{r_1 \dots r_n\}$ – obiekty (reguły) grupowane; t_{prec} - wartość sterująca momentem zakończenia grupowania;

Rezultat: $Tree[1], \dots, Tree[T]$ - T skupień reguł wraz z wyznaczonymi reprezentantami; T - optymalna liczba skupień

begin

Ustal parametry algorytmu;

Wyznacz macierz podobieństwa zgodnie z przyjętym kryterium odległości;

$simMax$:= maksymalna wartość podobieństwa dwóch reguł;

while *Wszystkie reguły nie są w jednym skupieniu* **do**

R_1, R_2 := Dwie najbardziej podobne reguły (skupienia) ;

Połącz R_1 oraz R_2 w jedno skupienie zgodnie z przyjętym kryterium ;

Zaktualizuj macierz podobieństwa;

end

$T := simMax \cdot t_{prec}$;

Przytnij dendrogram do uzyskania T grup;

Wyznacz reprezentantów każdego skupienia;

end

Dobór optymalnej wartości parametru t_{prec} stanowił przedmiot badań eksperymentalnych, których wyniki przedstawione są w kolejnym rozdziale. Algorytm $mAHC$ rozszerzony został o automatyczne wyznaczanie optymalnej liczby skupień w czasie grupowania. Dzięki temu nie jest konieczne wykonywanie oddzielnego przebiegu algorytmu w celu wyznaczenia wartości tego parametru.

Autor postanowił o kontynuowaniu procesu grupowania aż do momentu uzyskania pełnego dendrogramu, a następnie dopiero przycinaniu tak uzyskanego drzewa. Dzięki temu raz wygenerowana struktura hierarchiczna może być optymalizowana. Przycinanie drzewa na wysokim poziomie (czyli otrzymanie małej liczby grup) powoduje znaczne przyspieszenie procesu wyszukiwania reguł (skupień) relewantnych połączone jednak z ryzykiem utraty kompletności wyszukiwania. Przycięcie drzewa na niskim poziomie wiąże się z otrzymaniem dużej liczby małolicznych skupień, co umożliwi efektywniejsze wyszukiwanie reguł (skupień) relewantnych w zamian za zmniejszenie szybkości wyszukiwania.

Lista kroków algorytmu optymalnego doboru skupień:

1. Ustalenie parametrów grupowania: wybór miary podobieństwa, metody łączenia skupień, liczby skupień.
2. Rozpoczęcie algorytmu grupowania.
3. Zapamiętanie maksymalnej wartości podobieństwa dwóch reguł z pierwszego kroku algorytmu grupowania ($simMax$).
4. Wyznaczenie wartości progowej T współczynnika podobieństwa dwóch skupień będącej iloczynem $simMax$ oraz wartości t_{prec} podanej przez użytkownika określającej kiedy zakończyć grupowanie.

Jak zostanie to wykazane w dalszej części pracy (eksperyment opisany w rozdziale 8.4 na stronie 181), po przeprowadzeniu szeregu badań ustalono optymalną wartość współczynnika T wynoszącą $T = 0,85 \cdot simMax$.

5.3.4 Przegląd skupień reguł

Algorytm AHC generuje pełne drzewo reguł, co pozwala szybko wyszukiwać reguły porównując w każdym kroku zbiór faktów do reprezentantów lewego i prawego poddrzewa aż do zejścia do poziomu liści. Formalnie, jeśli przez D będziemy rozumieć zbiór deskryptorów (par atrybut-wartość), f_{sim}

jako funkcję podobieństwa, która dwóm regułom (skupieniom reguł) przyporządkowuje wartość podobieństwa, a przez k_i, l_i – węzły łączone, wtedy każda grupa w_i będzie definiowana jako $w_i = \{D_i, f_{sim}, k_i, l_i\}$, gdzie $D_i = \{d_1, \ldots, d_m\}, f_{sim} : U \times U \rightarrow \Re|_{[0...1]}$. Tego typu strukturę można przeszukiwać zgodnie z metodami zaproponowanymi w rozprawie:

a) metodą węzła najbardziej obiecującego,

b) metodą $mAHC$ z użyciem reprezentantów skupień.

Metoda węzła najbardziej obiecującego

Metoda węzła najbardziej obiecującego rozpoczyna wyszukiwanie od korzenia drzewa. Następnie w każdym kroku zbiór faktów Q porównywany jest z reprezentantami prawego i lewego poddrzewa aktualnie rozpatrywanego węzła, co przedstawia rys. 5.7. Do dalszej analizy wybierana jest ścieżka o większej wartości podobieństwa faktów do grupy reguł. Proces kończy się w momencie dotarcia do liścia oznaczającego konkretną regułę w bazie wiedzy lub też skupienia reguł w zależności od preferencji użytkownika. W przypadku wyboru skupienia, możliwe jest odnalezienie zbioru reguł najbardziej relewantnych w stosunku do aktualnego zbioru faktów.

Algorytm 7: Algorytm węzła najbardziej obiecującego

Dane: $Tree$ - Dendrogram zawierający skupienia reguł; Zbiór faktów Q
Rezultat: Skupienie reguł W relewantnych w stosunku do zbioru Q
begin
  W := Korzeń dendrogramu ;
  while *Nie osiągnięto założonej głębokości w drzewie* **do**
    $s_1 = f_{sim}(Q, Tree[L])$;
    $s_2 = f_{sim}(Q, Tree[R])$;
    if $s_1 >= s_2$ **then**
      $W := Tree[L]$;
    else
      $W := Tree[R]$;
    end
  end
end

Algorytm jest przystosowany do odnajdywania zarówno skupień reguł jak i samych reguł (liści w drzewie). Sterowanie głębokością wyszukiwania,

a co za tym idzie – licznością reguł otrzymanych w wyniku działania algorytmu, należy do użytkownika systemu poprzez ustawienie odpowiedniego parametru (szersze informacje znaleźć można w rozdziale 7.2). Możliwe jest wyszukiwanie aż do poziomu liści, wtedy w wyniku otrzymujemy dokładnie jedną regułę z bazy wiedzy. Możliwe jest również zatrzymanie wyszukiwania na wyższym poziomie i otrzymanie wyniku złożonego ze wszystkich reguł w aktualnie wybranym poddrzewie.

Problemem z przystosowaniem metody węzła najbardziej obiecującego jest wyznaczenie wartości podobieństwa zbioru faktów do poszczególnych węzłów. W tym celu autor proponuje trzy różne podejścia: *metodę pokrycia deskryptorowego, pokrycia atrybutowego* oraz *podejście hybrydowe.*

Najbardziej intuicyjną miarą podobieństwa dwóch zbiorów deskryptorów k, l jest wyznaczenie liczby deskryptorów występujących zarówno w zbiorze faktów, jak i w poszczególnych węzłach zgodnie ze wzorem:

$$f_{sim_d}(k, l) = card(d_k \cap d_l)$$

gdzie d_l oraz d_k to zbiory deskryptorów węzłów l i k odpowiednio.

Takie podejście, nazwane przez autora metodą pokrycia deskryptorowego, faworyzuje jednak węzły zawierające dużą liczbę powtarzających się, częstych deskryptorów w systemie. Co więcej, w kontekście wiedzy niepełnej, już informacja o wspólnych atrybutach występujących w obu grupach powinna być brana pod uwagę w wyliczaniu podobieństw (np. ze względu na niedoskonałości pomiaru, wartości puste, itp.). Zwłaszcza w bazach medycznych, często niepoddanych poprawnej dyskretyzacji, informacje o wykonaniu danego badania (bez znajomości wyniku) będą mogły być wykorzystane do dalszego rozróżniania grup pomiędzy sobą. Niestety, bez znajomości relacji pomiędzy wartościami poszczególnych atrybutów nie jest możliwe wyprowadzenie funkcji podobieństwa uwzględniającej które wartości atrybutów są sobie bliższe niż inne (tak jest dla większości atrybutów będących cechami nominalnymi [jakościowymi]). Dlatego też proponuje się ogólne podejście, właściwe niezależnie od rodzaju danych.

Drugim ze sposobów określania miary podobieństwa jest metoda pokrycia atrybutowego, która przy wyznaczaniu podobieństwa bierze pod uwagę tylko informacje o wspólnych atrybutach. Przy zachowaniu poprzedniej konwencji oznaczeń oraz gdy zbiory a_k oraz a_l oznaczać będą zbiory atrybutów reguł występujących w poszczególnych zbiorach deskryptorów, miara pokrycia atrybutowego będzie wyznaczana w sposób następujący:

$$f_{sim_a}(k,l) = card(a_k \cap a_l)$$

Ze względu na duże podobieństwo reguł do siebie oraz stosunkowo liczne zbiory wartości poszczególnych atrybutów, autor postanowił wykorzystywać tylko informacje o wspólnych atrybutach w obliczaniu podobieństwa dwóch węzłów. Dzięki temu można będzie wyróżnić, być może spójne, grupy reguł.

Trzecim zaproponowanym podejściem będzie połączenie dwóch poprzednich sposobów w metodzie hybrydowej:

$$f_{sim_h}(k,l) = card(d_k \cap d_l) \cdot B_1 + card(a_k \cap a_l) \cdot B_2$$

gdzie B_1 oraz B_2 to współczynniki stopniujące takie, że $B_1,\ B_2 \in [0 \ldots 1]$. Wartość współczynników B_1 oraz B_2 wyznaczana jest eksperymentalnie, co przedstawia przykład poniżej oraz eksperyment przedstawiony w rozdziale 8.5.

Jak pokazały eksperymenty, podejście to wykorzystuje zalety zarówno dokładności metody z pokryciem deskryptorów, jak również istnienie dodatkowych informacji rozróżniających reguły pomiędzy sobą. Współczynniki stopniujące służą zwiększaniu lub zmniejszaniu ważności części deskryptorowej i atrybutowej. W rozważaniach sprawdzono dwa przypadki, w jednym znacznie większą ważność otrzymuje część deskryptorowa, w drugim – część atrybutowa.

Aby zobrazować sposób wyliczania powyższych wartości, przeanalizujmy przykład. Dla dwóch węzłów k i l: $d_k = \{(a=1),(a=1),(a=2),(b=1),(b=1),(c=1)\}$, $d_l = \{(a=2),(a=2),(a=1),(b=1),(b=1),(c=1)\}$ i zbioru faktów $Q = \{(a=2),(c=1)\}$ odpowiednie wartości podobieństw przedstawiają się następująco:

- $f_{sim_d}(k,Q) = 2$; $f_{sim_d}(l,Q) = 3$,
- $f_{sim_a}(k,Q) = 4$; $f_{sim_a}(l,Q) = 3$,
- dla $B_1 = 0,75$ oraz $B_2 = 0,25$ $f_{sim_h1}(k,Q) = 2,5$; $f_{sim_h2}(l,Q) = 3$,
- dla $B_1 = 0,25$ oraz $B_2 = 0,75$ $f_{sim_h2}(k,Q) = 3,5$; $f_{sim_h2}(l,Q) = 3$.

Widać wyraźnie, że podejście hybrydowe pozwala na uwzględnienie również współwystępowania atrybutów w zbiorze faktów oraz analizowanej regule i przyczyni się do wyboru optymalnej ścieżki w warunkach dużego podobieństwa reprezentantów węzłów obu analizowanych ścieżek.

Wyszukiwanie z użyciem reprezentantów skupień

Alternatywą do podejścia korzystającego z metody węzła najbardziej obiecującego jest skorzystanie z algorytmu $mAHC$ wyznaczającego reprezentantów poszczególnych skupień i porównując z nimi zbiór faktów. Takie podejście również przyspiesza odnajdywanie reguł relewantnych, jednakże jest silnie uzależnione od momentu przycięcia dendrogramu[§]. Jeśli grup będzie stosunkowo dużo, zysk czasowy w czasie odnajdywania reguł możliwych do uaktywnienia będzie znikomy.

Algorytm do swojego działania wymaga k reprezentantów skupień. Po ich wyznaczeniu, następuje porównanie zbioru faktów do każdego z nich, a w wyniku zwracane jest skupienie, którego reprezentant jest najbardziej podobny.

Należy zauważyć, że analogicznie w stosunku do poprzedniego rozwiązania, wyznaczenie reprezentantów jest czynnością jednorazową. Każdorazowe późniejsze wyszukiwanie może odbywać się dzięki temu znacznie szybciej.

Algorytm 8: Algorytm wyszukiwania z użyciem reprezentantów skupień

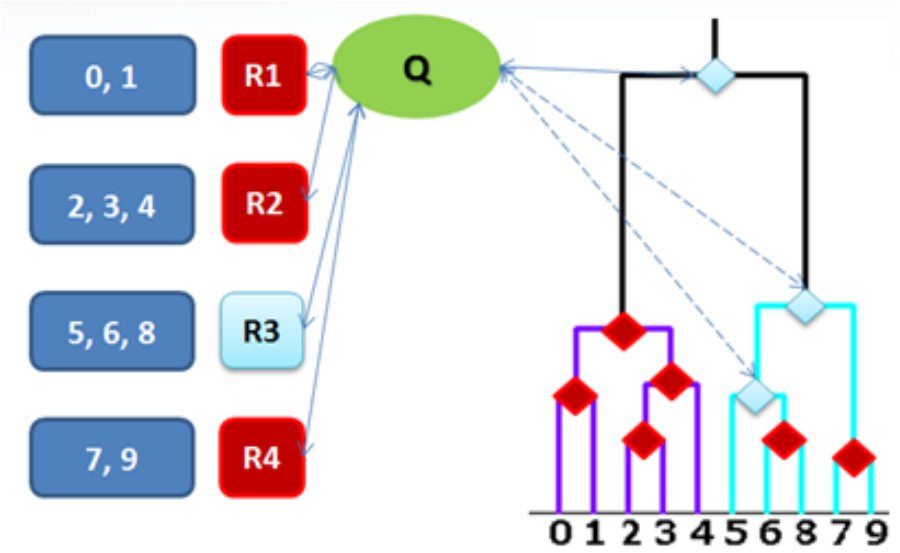
Dane: $Tree[1], \ldots, Tree[k]$ - k skupień reguł wraz z wyznaczonymi reprezentantami; Zbiór faktów Q
Rezultat: Skupienie reguł W relewantnych w stosunku do zbioru Q

```
begin
    W := Tree[1] ;
    for i ← 2 to k do
        if f_sim(W < f_sim(Tree[i])) then
            W := Tree[i] ;
        end
    end
end
```

Rysunek 5.7 na stronie 129 przedstawia różnice pomiędzy wyszukiwaniem z użyciem skupień reguł i wyszukiwaniem w strukturze hierarchicznej. Pierwsze z nich wykorzystuje wyznaczonych wcześniej reprezentantów skupień $R1 \ldots R4$ w celu porównania ich z aktualnie rozpatrywanym zbiorem faktów. W wyniku zwrócona jest grupa posiadająca największe podobieństwo reprezentanta do zbioru faktów (w przykładzie $R3$). Wyszukiwanie z użyciem hierarchii korzysta z metody węzła najbardziej obiecującego aby na każdym poziomie hierarchii porównywać aktualny zbiór faktów do repre-

[§]Innymi słowy, od liczby utworzonych grup

zentantów lewego i prawego poddrzewa. W wyniku procesu wyszukiwania hierarchicznego, zwracana jest grupa reguł (lub w szczególnym przypadku pojedyncza reguła) najbardziej podobnych do aktualnego zbioru faktów. W przykładzie jest to grupa złożona z reguł 5, 6 oraz 8.



Rysunek 5.7: Wyszukiwanie z użyciem skupień reguł (z lewej) i wyszukiwanie hierarchicznym (z prawej)

Szczegółowe eksperymenty porównujące oba te podejścia znajdują się w dalszej części rozprawy.

5.3.5 Metoda współczynników IF

System po odnalezieniu grupy najbardziej podobnej do aktualnego zbioru faktów, przystępuje do analizy reguł wchodzących w skład tej grupy. Jeśli znajdują się tam reguły mające pełne pokrycie w zbiorze faktów, te zostają uaktywniane. W przypadku, gdy nie ma reguł mających pełne pokrycie w zbiorze faktów, uaktywniane są reguły mające tylko częściowe pokrycie. W celu odróżnienia wiedzy wyznaczonej przez reguły pewne od reguł niepewnych wprowadza się współczynnik niepełności IF (ang. *incompleteness factor*). Współczynnik ten będzie wyznaczany w następujący sposób:

$$IF(d_i) = \frac{\sum IF(f_j)}{card(D_i)}; f_j \in (F \cap D_i)$$

gdzie $IF(d_i)$ to współczynnik niepełności i-tej przesłanki, $IF(f_j)$ to współczynnik niepełności j-tego faktu będącego częścią wspólną zbioru

deskryptorów D_i oraz zbioru faktów (F). Inaczej mówiąc, przez f_j oznaczać będziemy deskryptory wchodzącego w skład reguły i mające pokrycie w zbiorze faktów. W trakcie procesu wnioskowania część faktów będzie niepewna, bo są one konkluzją niepełnych reguł. Miarą tej niepełności będzie zaproponowany współczynnik IF.

Fakty pewne, znane wcześniej lub też dopisane do zbioru faktów po uaktywnieniu reguł pewnych posiadają z definicji wartość współczynnika IF równą 1. Współczynnik IF ma wartości z przedziału $[0 \ldots 1]$, co pozwala na łatwą interpretacje uzyskanych wyników. Przykładowo, dla zbioru faktów $\{(a = 1), (b = 2), (c = 5)\}$ oraz następujących reguł:

R1: (a=1) & (b=2) & (c=3) $\rightarrow$ (d=1),

R2: (a=1) & (b=3) & (c=4) $\rightarrow$ (d=2),

współczynniki IF tychże reguł wynoszą odpowiednio:

1. $IF(R1) = 0,67$,
2. $IF(R2) = 0,33$.

Autor proponuje również wprowadzenie średniej wartości współczynnika IF dla całej odnalezionej grupy. W przedstawianym przykładzie wartość tego współczynnika dla grupy złożonej z reguł R_1 oraz R_2 wynosiłaby 0,5.

Współczynnik IF powstał dzięki inspiracji współczynnikami CF zaproponowanymi w systemie MYCIN [152].

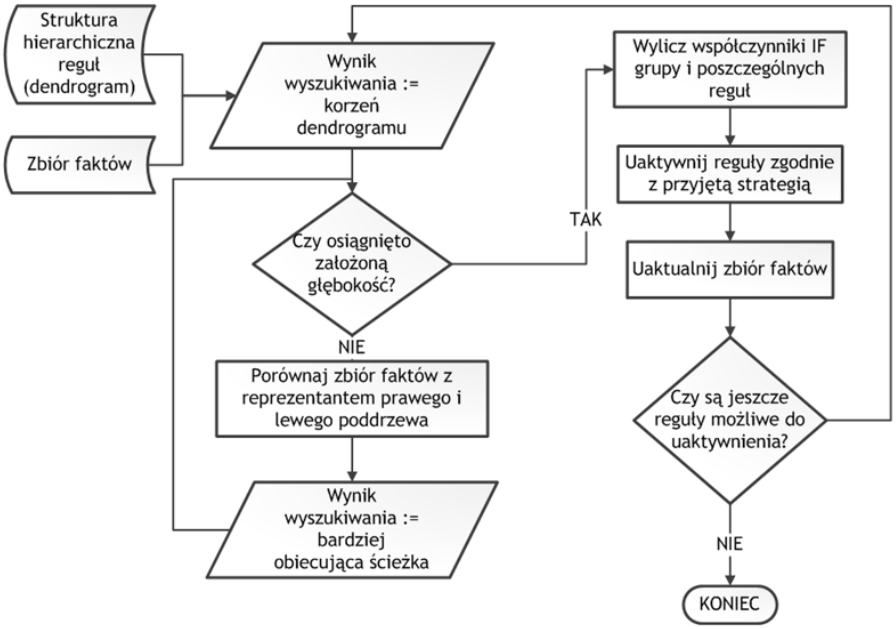
Współczynnik IF zaproponowany przez autora wykorzystuje jeszcze prostszy obliczeniowo sposób określania niepełności wiedzy. Stanowi łatwą miarę do modelowania niepewności i niepełności wiedzy zależną tylko od stopnia pełności wiedzy zapisanej w zbiorze faktów. Umożliwia korzystanie z zalet mechanizmów grupowania (m.in. wybór grupy o największym współczynniku IF). Umożliwia się również sterowanie "jakością" wnioskowania. Do użytkownika systemu należy decyzja, czy liczy się z możliwością otrzymania wiedzy niższej jakości, czy woli nie otrzymać żadnych nowych (niepełnych) informacji w wyniku procesu wnioskowania.

Wnioskowanie w warunkach wiedzy niepełnej

Cały proces wnioskowania w warunkach wiedzy niepełnej przedstawiony jest schematycznie na rys. 5.8 oraz zaprezentowany w postaci pseudokodu na stronie 132.

Jak widać, proponowane rozwiązanie do poprawnego działania wymaga zbioru faktów wraz z uzupełnionymi wartościami współczynników IF, a także bazy wiedzy danej w postaci dendrogramu. System na początku rozpoczyna wyszukiwanie skupienia reguł najbardziej relewantnego w stosunku do zbioru faktów, po jego znalezieniu następuje uaktywnienie reguł w nim występujących zgodnie z przyjętą strategią aktywowania reguł. Uaktualniony zostaje zbiór faktów, co pozwala na dalsze wnioskowanie.

Parametrem danym przez użytkownika jest po pierwsze głębokość wyszukiwania w dendrogramie (a co za tym idzie przybliżona liczność otrzymanej grupy) oraz minimalna progowa wartość współczynnika IF poniżej której fakty traktowane są jako niepewne.



Rysunek 5.8: Wyszukiwanie w hierarchicznej strukturze reguł

Algorytm 9: Metoda współczynników IF do wnioskowania w warunkach wiedzy niepełnej

Dane: $U = \{r_1 \dots r_n\}$ – płaska struktura reguł; Zbiór faktów Q wraz z wartościami współczynników IF
Rezultat: Uaktualniony zbiór faktów
begin
  `/* Moduł grupujący */`
  Ustal parametry algorytmu grupującego;
  Wyznacz macierz podobieństwa zgodnie z przyjętym kryterium odległości;
  while *Wszystkie reguły nie są w jednym skupieniu* **do**
    R_1, R_2 := Dwie najbardziej podobne reguły (skupienia) ;
    Połącz R_1 oraz R_2 w jedno skupienie zgodnie z przyjętym kryterium ;
    Zaktualizuj macierz podobieństwa;
  end
  repeat
    `/* Moduł wyszukujący reguły */`
    W := Korzeń dendrogramu ;
    while *Nie osiągnięto założonej głębokości w drzewie* **do**
      $s_1 = f_{sim}(Q, Tree[L])$;
      $s_2 = f_{sim}(Q, Tree[R])$;
      if $s_1 >= s_2$ **then**
        $W := Tree[L]$;
      else
        $W := Tree[R]$;
      end
      `/* Moduł wnioskowania */`
      Wylicz współczynniki IF poszczególnych reguł wchodzących w skład skupienia W ;
      Uaktywnij regułę R ze skupienia W zgodnie z przyjętą strategią;
      Uaktualnij zbiór faktów ;
    end
  until *Są jeszcze reguły możliwe do uaktywnienia*;
end

Przykład

Korzystając z przykładowej wiedzy dziedzinowej zapisanej w rozdziale 4.2 na stronie 53 pokazany zostanie sposób działania prezentowanego systemu.

Na początku, podobnie jak w przypadku logiki rozmytej, należy dokonać syntezy wiedzy do postaci regułowej tożsamej do przedstawionej w rozdziale 4.5 na stronie 64.

Wiedzę tą można również zapisać w postaci tablic decyzyjnych (rys. 5.8 oraz 5.9 na stronie 133).

X	Stan roweru (S)	Intensywność treningu (I)	Wypadek na trasie (W)	Wynik maratonu (M)
X_1	dobry	odpowiednia	tak	wygrana
X_2	dobry	odpowiednia	nie	wygrana
X_3	dobry	nieodpowiednia	tak	przegrana
X_4	dobry	nieodpowiednia	nie	podium
X_5	zły	odpowiednia	tak	przegrana
X_6	zły	odpowiednia	nie	podium
X_7	zły	nieodpowiednia	tak	przegrana
X_8	zły	nieodpowiednia	nie	podium

Tabela 5.8: Tablica decyzyjna dla przykładowej wiedzy

X	Wynik maratonu (M)	Humor sponsora (H)	Premia (P)
R_1	wygrana	dobry	przyznana
R_2	wygrana	zły	przyznana
R_3	podium	dobry	przyznana
R_4	podium	zły	nie przyznana
R_5	przegrana	dobry	nie przyznana
R_6	przegrana	zły	nie przyznana

Tabela 5.9: Druga tablica decyzyjna dla przykładowej wiedzy

System do efektywnego działania powinien pracować na wygenerowanych regułach minimalnych, stąd wyznaczone reguły minimalne będą miały postać:

R1: JEŻELI (S=dobry) ORAZ (I=odpowiednia) TO (M=wygrana)

R2: JEŻELI (I=nieodpowiednia) ORAZ (W=tak) TO (M=przegrana)

R3: JEŻELI (I=nieodpowiednia) ORAZ (W=nie) TO (M=podium)

R4: JEŻELI (S=zły) ORAZ (W=tak) TO (M=przegrana)

R5: JEŻELI (S=zły) ORAZ (W=nie) TO (M=podium)

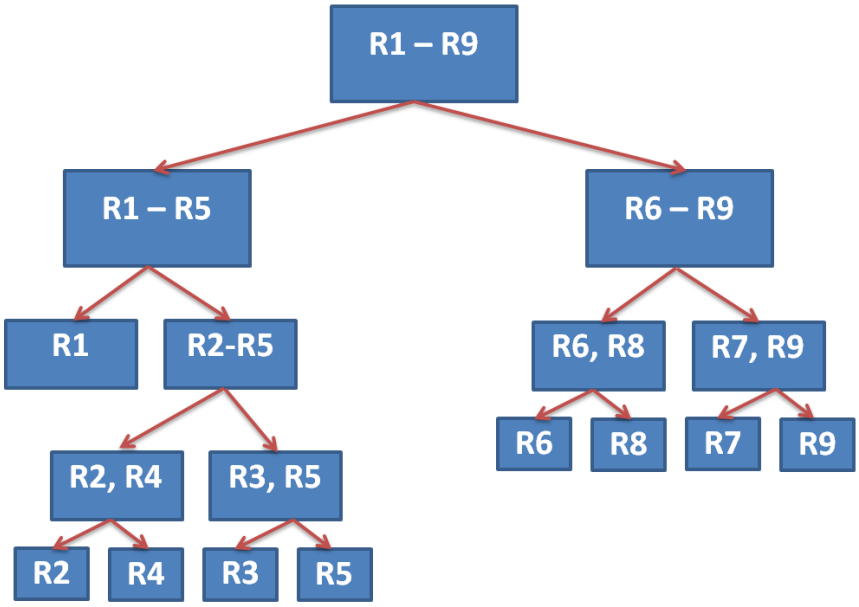
R6: JEŻELI (M=wygrana) TO (P=przyznana)

R7: JEŻELI (M=przegrana) TO (P=nie)

R8: JEŻELI (M=podium) ORAZ (H=dobry) TO (P=przyznana)

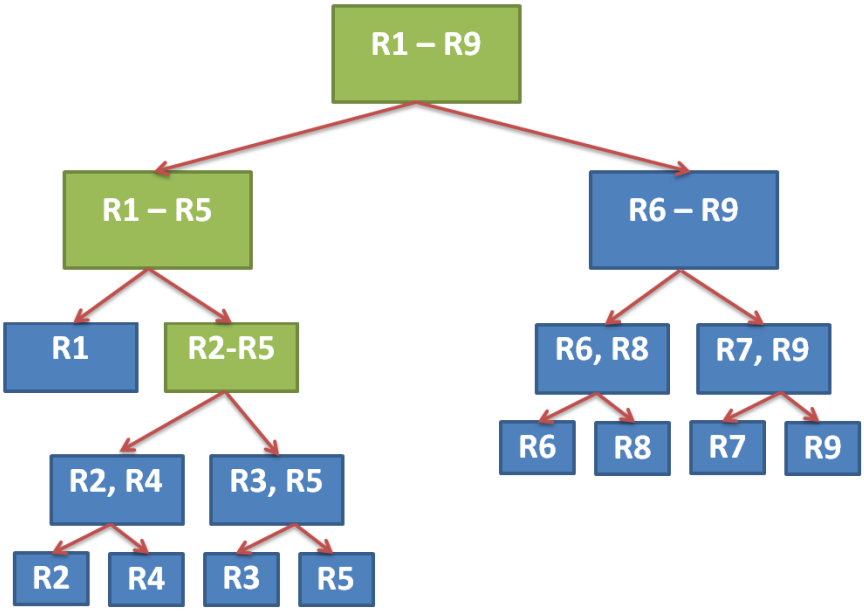
R9: JEŻELI (M=podium) ORAZ (H=zły) TO (P=nie)

W pierwszym kroku należy wygenerować skupienia z przedstawionych reguł. Przykładowe skupienia wygenerowane przez algorytm *AHC* przedstawia rysunek 5.9:



Rysunek 5.9: Dendrogram dla przykładowego zestawu reguł

Zakładając, że zbiór faktów składa się z dwóch deskryptorów: {(I=nieodpowiednia), (S=zły)} wtedy korzystając z metody węzła najbardziej obiecującego proces wnioskowania będzie zaznaczony zielonym kolorem na rys. 5.10. W przykładzie zakładamy głębokość wyszukiwania równą 2.



Rysunek 5.10: Ścieżka wnioskowania dla przykładowych danych

W tym momencie przystępujemy do faktycznego wnioskowania uaktywniając reguły wchodzące w skład tego skupienia i wyliczając wartość współczynnika IF dla każdej z nich:

R2: JEŻELI (I=nieodpowiednia) ORAZ (W=tak) TO (M=przegrana) $\Rightarrow$ $CF(R1) = 0,33$

R3: JEŻELI (I=nieodpowiednia) ORAZ (W=nie) TO (M=podium) $\Rightarrow$ $CF(R2) = 0,33$

R4: JEŻELI (S=zły) ORAZ (W=tak) TO (M=przegrana) $\Rightarrow$ $CF(R3) = 0,33$

R5: JEŻELI (S=zły) ORAZ (W=nie) TO (M=podium) $\Rightarrow$ $CF(R4) = 0,33$

Użytkownik w wyniku wnioskowania otrzymuje informację, że w bazie nie ma reguł, które w pełnym stopniu odpowiadają zakładanemu zbiorowi faktów. Jednocześnie prezentowana jest grupa zawierająca reguły, które są pokryte zbiorem faktów w największym możliwym stopniu. To od użytkownika zależy teraz, czy pozwolić na uaktywnienie tychże reguł i dopuścić do dalszej propagacji niepewności wiedzy.

Dodatkowym atutem przedstawionego podejścia jest jednoczesne przedstawienie przesłanek koniecznych do udowodnienia w celu otrzymania reguł w pełni pokrytych zbiorem faktów. Dla przedstawionego przykładu istotna byłaby wartość atrybutu "W" która pozwoliłaby na uaktywnienie jednej z odnalezionych reguł bez niepewności wiedzy.

5.4 Podsumowanie

W pierwszych zastosowaniach grupowanie było stosowane do danych. Potem przeniesiono te mechanizmy do grupowania dokumentów w systemie SMART Saltona. Ideą jego działania było grupowanie dokumentów na podstawie ich podobieństwa między sobą [135]. Podobnie jak w proponowanym podejściu, wyszukiwanie odbywało się w znacznie krótszym czasie w stosunku do przeszukiwania liniowego, ze względu na wykorzystanie reprezentantów grup i struktury drzewiastej. W rozprawie proponuje się zastosowanie algorytmu grupującego AHC do stworzenia skupień złożonych z reguł tworzących bazę wiedzy. Poza znacznym przyspieszeniem wyszukiwania, uzyskuje się również informacje o grupach najbardziej podobnych reguł. Ta hierarchiczna struktura bazy reguł pozwoli w dalszym etapie na odnalezienie grupy reguł najbardziej podobnych do aktualnego zbioru faktów. Dzięki temu, w sytuacji w której żadna z reguł nie może zostać uaktywniona, możliwym będzie uaktywnienie reguł zbliżonych przy jednoczesnym oznaczeniu wiedzy przez nie generowanej jako niepewnej. Jedyną zauważalną wadą algorytmu jest jego relatywnie wysoka złożoność obliczeniowa zależna od liczby reguł w systemie [rzędu $O(n^2)$]. Do zalet z całą pewnością należy zaliczyć odporność na wartości odstające, jak i fakt, że przeszukiwanie drzewa odbywa się w czasie logarytmicznym – złożoność tego procesu to $O(log_2 n)$ typowa dla drzew binarnych.

Proponowane podejście różni się od modelu klasycznego SWD tym, że przechowuje reguły w strukturze drzewiastej oraz określa podobieństwo tych reguł w stosunku do aktualnego zbioru faktów na każdym etapie wnioskowania. Wyszukiwanie odbywa się za pomocą metody węzła najbardziej obiecującego. Została ona zmodyfikowana i przystosowana do działania dla regułowych baz wiedzy.

Postanowiono wykorzystać algorytm AHC pomimo jego relatywnie dużej złożoności obliczeniowej. Ogromną zaletą wobec algorytmów niehierarchicznych jest jednak fakt utworzenia struktury hierarchicznej pozwalającej na

szybkie wyszukiwanie reguł, a co za tym idzie – zwiększenie efektywności wnioskowania.

Niepełność wiedzy jest problemem nietrywialnym. Proponowane przez autora podejście łączy metody znane z analizy skupień, elementów teorii zbiorów przybliżonych oraz współczynników pewności CF w celu łatwiejszego i wydajniejszego modelowania niepewności wiedzy. W przedstawionym przykładzie zilustrowano przydatność i celowość tego rozwiązania. Należy zauważyć zysk czasowy z wprowadzenia tego podejścia: zamiast płaskiego przeszukiwania w bazie złożonej z 9 reguł, należało wykonać tylko 2 porównania zbioru faktów i reprezentantów skupień na poszczególnych poziomach hierarchii. W wyniku działania systemu nie tylko otrzymaliśmy grupę reguł najbardziej relewantnych w stosunku do zbioru faktów, ale również informację o stopniu pokrycia zbiorem faktów tychże reguł.

Rozdział 6

Ocena efektywności

Efektywność jest pojęciem używanym nie tylko do systemów informatycznych. Spotykamy się z nim na co dzień, przykładowo mówiąc o efektywnym zarządzaniu czasem. Słownik języka polskiego PWN za "efektywny" uznaje:

1. dający dobre wyniki; wydajny;
2. istotny, rzeczywisty.

Każdy system informatyczny jest tworzony z myślą o wydajnym i poprawnym działaniu. Aby ocenić te trudnomierzalne aspekty, powstały liczne sposoby oceny jakości i efektywności systemów informatycznych. Metody oceny służą także do porównywania i analizy wyników uzyskiwanych przez konkurencyjne algorytmy i systemy. Przedstawiany w rozprawie sposób optymalizacji procesów wnioskowania realizuje także proces grupowania reguł i reprezentację wiedzy niepełnej, niezbędne wydaje się przeprowadzenie oceny efektywności jego działania. Problem oceny efektywności jest bardzo szeroki i przekracza ramy tej rozprawy. Wyczerpujące informacje nt. metod oceny jakości systemów wspomagania decyzji znaleźć można w pozycjach [41, 43, 46], natomiast studium oceny jakości grupowania przedstawione zostało w [17, 29, 40, 56–58, 82, 150, 169].

Proponowany system będzie oceniany w trzech kategoriach:

1. Ocena jakości tworzonej struktury skupień reguł w bazie wiedzy (innymi słowy: efektywność bazy wiedzy). Wykorzystane tu będą miary oceny jakości skupień.

2. Ocena jakości procesu wnioskowania w systemach ekspertowych. Badane i oceniane będą tu zarówno efektywność czasowa jak i pamięciowa wyszukiwania reguł do uaktywnienia.

3. Ocena efektywności wnioskowania z wiedzą niepełną z wykorzystaniem metody współczynników IF. W tym kontekście pokazane zostaną zalety i zyski proponowanej metody wnioskowania.

6.1 Ocena jakości struktury skupień reguł

W literaturze [17, 29, 40, 56–58, 82, 150, 169] spotkać można wiele metod służących ocenie poprawności grupowania. Cel oceny jest jasny – po przeprowadzeniu analizy skupień otrzymujemy pewien podział obiektów na grupy. W celu porównania działania kilku algorytmów (lub też oceny wpływu zmiany parametrów jednego algorytmu) należy porównać ze sobą wyniki grupowania używając jednej z miar zaprezentowanych poniżej.

Zdarza się tak, że nawet drobna zmiana parametru wyraźnie zmienia jakość i sposób grupowania danych (np. w przypadku podejść ewolucyjnych do grupowania danych). Przykładowo, gdy w zbiorze poddawanym grupowaniu występują naturalnie dwa skupienia, a wymuszamy tworzenie trzech bądź większej liczby grup, wtedy musimy się liczyć z tym, że jakość grupowania nie będzie optymalna.

W wielu opracowaniach, m.in. [57] autorzy przyjmują podział metod oceny na trzy podgrupy:

1. wewnętrzne,

2. zewnętrzne,

3. względne.

Metody wewnętrzne nie potrzebują do przeprowadzenia skutecznej oceny żadnych dodatkowych danych z zewnątrz. Mowa tu o dodatkowej wiedzy dostarczonej np. przez eksperta oceniającego jakość grupowania. Są przykładem metod nienadzorowanych, albowiem potrafią działać w sposób całkowicie automatyczny [154]. Metody wewnętrzne zwykle opierają się na ocenie jednego z dwóch parametrów: miary separacji wyznaczającej stopień oddzielenia poszczególnych grup od siebie oraz miary spójności, obliczające stopień jednorodności danej grupy.

Metody zewnętrzne z kolei korzystają z wcześniej ustalonego optymalnego podziału obiektów na skupienia. W tym celu należy dostarczyć wiedzy

z zewnątrz aby algorytm mógł porównać rezultaty grupowania z wcześniej ustalonym wzorcem.

Metody względne są zwykle wykorzystywane do oceny jakości grupowania w obrębie zmiany parametrów tego samego algorytmu lub też kilku algorytmów działających na tych samych danych. Ich nazwa odzwierciedla zachowanie – obliczają względną jakość grupowania.

Aby zmierzyć jakość grupowania, wprowadzić można pojęcie *całkowitej poprawności* rozumianej jako suma ważona poprawności poszczególnych grup $C_1 \ldots C_i \ldots C_K$:

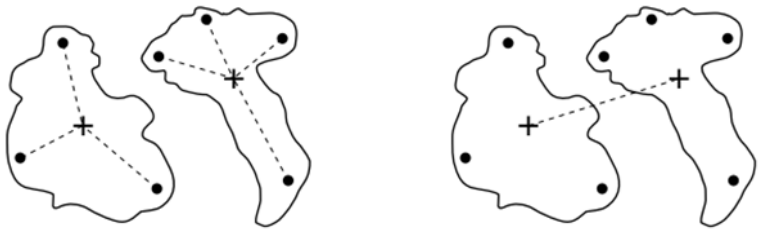
$$\text{całkowita poprawność } = \sum_{i=1}^{K} w_i \cdot \text{ poprawność}(C_i).$$

Poprawność cząstkowa C_i może być dowolną z miar poprawności poszczególnych klastrów. W wersji klasycznej, wszystkie wagi w_i są równe 1, jednakże istnieje możliwość zmiany wag dla poszczególnych grup.

6.1.1 Ocena wewnętrzna

Ocena wewnętrzna opiera się tylko na danych przeznaczonych do grupowania. Przykładem takiej oceny mogą być miary separacji (ang. *separation*) oraz spójności (ang. *compactness, cohesion*) skupienia [18]. Zwykle są one stosowane do algorytmów z grupy niehierarchicznych.

Miary spójności i separacji



Rysunek 6.1: Graficzna interpretacja pojęcia spójności (po lewej) i separacji (po prawej)

Definicja miar *spójności* i *separacji* jest intuicyjna (patrz graficzna reprezentacja na rys. 6.1): im bardziej homogeniczna grupa, tym miara jej spójności będzie większa. Im wyraźniej odróżnione grupy od siebie, tym

współczynnik separacji większy. Do mierzenia tych pojęć używa się różnych miar, np. sumy kwadratów odległości wewnątrz skupienia dla miary spójności lub sumy kwadratów odległości pomiędzy skupieniami dla miary separacji.

Suma kwadratów błędów

Suma kwadratów błędów (ang. *sum of squared errors - SSE*) jest podejściem znanym ze statystyki. W przypadku analizy skupień, suma kwadratów błędów wyliczana jest jako suma odległości każdego obiektu do tzw. prototypu. Prototypem może być obiekt uznany wcześniej za najlepszego reprezentanta skupienia albo wybrany metodami obliczeniowymi. Miarę tę można zdefiniować następująco:

$$SSE = \sum_{i=1}^{K} \sum_{r \in C_i} dist(c_i, r)^2$$

gdzie K to liczba skupień, C_i oznacza i-te skupienie, a funkcja $dist(c_i, r)$ to odległość obiektu (reguły) r do prototypu grupy c_i.

SSE jest całkowicie nieodporny na istnienie wartości odstających (izolowanych). Współczynnik ten może być stosowany do przybliżonego określania, czy wybrana liczba skupień K jest optymalna dla zestawu danych poprzez wyliczanie wartości miary $eSSE$ w następujący sposób:

$$eSSE = e^{\frac{-SSE}{K}}.$$

Poprawny dobór liczby grup w stosunku do naturalnych skupień ma miejsce, gdy wartość $eSSE$ jest bliska 0. Niestety, współczynnik ten jest nieodporny na obiekty izolowane zawarte w skupieniach. Z tego powodu nie będzie brany pod uwagę w ocenie zaproponowanego w rozprawie systemu.

Współczynnik sylwetki

Kombinacją miar spójności i separacji jest tzw. współczynnik sylwetki (ang. *silhouette coefficient*) [76, 154].

Niech $a(i)$ będzie średnią odległością obiektu od wszystkich pozostałych elementów znajdujących się w obrębie tego samego skupienia. Dla każdej grupy C_m, w której nie znajduje się obiekt i wyznacz średnią odległość pomiędzy wszystkimi obiektami zawartymi w tej grupie a i-tym obiektem. Za $b(i)$ przyjmij minimalną wartość:

$$b(i) = min\{\forall_{C_m \cap i = \emptyset} \forall_{x \in C_m} \frac{\sum dist(i,x)}{card(C_m)}\}.$$

Wtedy współczynnik sylwetki $s(i)$ przyjmuje wartość:

$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}}.$$

Współczynnik sylwetki przyjmuje wartości z przedziału $[-1 \ldots 1]$. Jeżeli $s(i) < 0$, wtedy mówimy, że obiekt i został błędnie zgrupowany. Jeśli $s(i) \approx 0$ nie wiadomo, czy obiekt i jest przydzielony do poprawnej grupy, wreszcie gdy $s(i) > 0$ oznacza prawidłowe przyporządkowanie. Przyjmuje się również, że wartości $s(i) < 0,5$ oznaczają słabą jakość utworzonych grup.

Współczynnik sylwetki obliczany jest dla każdego obiektu oddzielnie co sprawia, że dla dużych zbiorów danych miara ta staje się nieefektywna (zbyt długi czas potrzebny na jej wyznaczenie). Wartości współczynnika sylwetki w obrębie grupy można uśrednić wyliczając średnią wartość współczynnika sylwetki (S_j) dla j -tej grupy zgodnie ze wzorem:

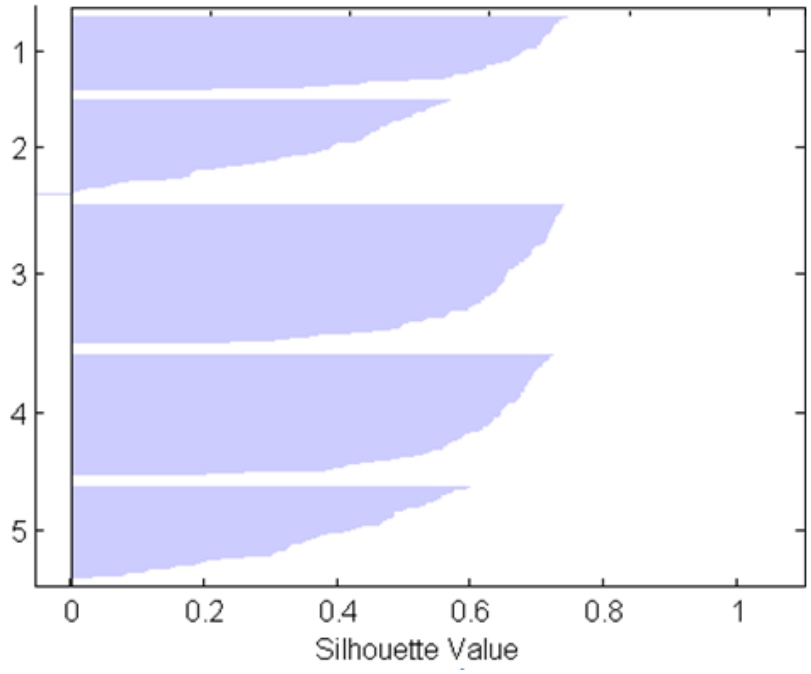
$$S_j = \frac{1}{card(C_j)} \sum_{i \in C_j} s(i)$$

Dalsze uogólnienie współczynnika sylwetki to tzw. całkowity (lub ogólny, globalny) współczynnik sylwetki GS_U (ang. *global silhouette*) dla U-tego podziału K skupień:

$$GS_U = \frac{1}{K} \sum_{j=1}^{K} S_j.$$

GS_U jest wykorzystywany do szacowania optymalnej liczby grup na które należy podzielić dane. Gdy w wyniku działania dowolnego algorytmu otrzymamy kilka różnych podziałów, należy wyliczyć współczynnik GS dla każdego z podziałów U, a następnie wybrać ten podział, którego GS jest największy.

Interpretacja graficzna współczynnika sylwetki może być podobna do rysunku 6.2. Na osi OX zaznaczone są wartości współczynnika sylwetki dla każdej z utworzonych grup oznaczonych etykietami jak na osi OY. Im rysunek bardziej przypomina foremne prostokąty, tym lepszy jakościowo podział. Wykres na którym dowolne skupienie szybko obniża swój współczynnik sylwetki obrazuje złą jakość procesu grupowania (patrz skupienie nr 5).



Rysunek 6.2: Graficzna interpretacja globalnego współczynnika sylwetki. Źródło: program MatLab

Indeks Dunna

Dla wyraźnie rozgraniczonych skupień, możliwe jest skorzystanie z indeksu Dunna [58]. W celu jego zdefiniowania, należy wcześniej przybliżyć pojęcia minimalnej odległości wewnętrznej skupienia D_{min} oraz maksymalnej odległości wewnętrznej skupienia (średnicy) $\Delta(C_i)$:

$$D_{min}(C_i, C_j) = min_{x \in C_i, y \in C_j; C_i \neq C_j}\{dist(x, y)\} \tag{6.1}$$

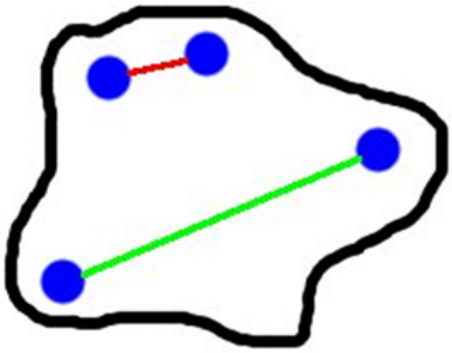
$$\Delta(C_i) = max_{x,y \in C_i} dist(x, y). \tag{6.2}$$

Interpretacja graficzna tych dwóch pojęć, zakładająca przyjęcie euklidesowej miary odległości, przedstawiona została na rysunku 6.3.

Wreszcie, indeks Dunna jest dany wzorem:

$$ID = \min_{1 \leqslant i \leqslant K}\{\min_{\substack{1 \leqslant j \leqslant K; \\ j \neq i}}\{\frac{D_{min}(C_i, C_j)}{\max_{j \leqslant m \leqslant K} \Delta(C_m)}\}\}.$$

Innymi słowy, indeks Dunna to iloraz z minimalnej wartości odległości wewnętrznej skupienia dwóch dowolnych obiektów podzielony przez największą średnicą dowolnej grupy. Indeks ten z definicji przyjmuje tylko war-



Rysunek 6.3: Ilustracja maksymalnego (kolor zielony) i minimalnego (kolor czerwony) dystansu wewnątrz grupy

tości dodatnie, nie ma natomiast górnego limitu wartości. Wyższa wartość oznacza lepszą jakość grupowania.

Indeks Dunna jest również miarą obciążoną przez występowanie wartości izolowanych, które powodują znaczne zwiększenie się średnicy grupy. Istotną wadą tego wskaźnika jest również stosunkowo duża złożoność obliczeniowa (rzędu wielomianowego).

6.1.2 Ocena zewnętrzna

Ocena jakości grupowania bez dodatkowych informacji z zewnątrz jest jedną z metod oceny jakości. W przypadku, gdy *a priori* dany jest jednak tzw. optymalny podział (czyli podział obiektów na grupy uznany za optymalny) możliwe jest zastosowanie metod oceny z grupy zewnętrznych. Optymalny podział może być dany np. za sprawą eksperta (lekarz oceniający przypadki chorobowe) lub też znany powszechnie (np. organizmy żywe i ich uprzednia taksologia).

Mając daną najlepszą (optymalną) klasyfikację można w stosunkowo łatwy sposób porównać ją z klasyfikacją dokonaną automatycznie (i tym samym odpowiedzieć na pytanie, czy tak postawiony problem da się zautomatyzować). W dalszej części rozdziału przyjmuje się, że optymalna klasyfikacja operuje na pojęciu klas: obiekt należy do danej klasy, co oznacza, że w przypadku idealnym algorytm grupujący powinien przyporządkować ten dokument do grupy tożsamej z tą klasą.

W obrębie oceny zewnętrznej autorzy [154] wyróżniają dwa podejścia: klasyfikacyjne oraz z użyciem miar podobieństwa. Pierwsze z nich opiera się

na mierzeniu stopnia zawierania obiektów jednej klasy przez poszczególne grupy (homogeniczność), drugie natomiast sprawdza w jak dużym stopniu obiekty etykietowane tą samą klasą zawierają się w tej samej grupie i odwrotnie.

Miary zorientowane na klasyfikację

Mierzona jest tutaj zgodność przyporządkowania do odpowiednich grup na podstawie znanych wcześniej klas [40].

Najczęściej wyróżniamy następujące miary:

Entropia (ang. *entropy*) jest miarą określającą jaka liczba obiektów danej klasy znajduje się w skupieniu. Inaczej mówiąc, jest to obliczenie stopnia dystrybucji klas w danej grupie. Entropia $i-$tego skupienia e_i po podziale na K grup dana jest wzorem:

$$e_i = -\sum_{j=1}^{K} p_{ij} \log_2 p_{ij}$$

gdzie

$$p_{ij} = \frac{m_{ij}}{m_i}$$

oznacza prawdopodobieństwo napotkania obiektu klasy j w grupie i wyznaczane jako iloraz liczby m_{ij} obiektów klasy j w grupie i do liczności $i$tej grupy oznaczanej m_i.

Entropia poszczególnych grup po zsumowaniu wraz z czynnikami wagowymi będącymi ilorazem liczności $i-$tej grupy m_i i liczności zbioru danych N przyjmuje postać całkowitej miary entropii:

$$e = \sum_{i}^{K} \frac{m_i}{N} \cdot e_i.$$

Współczynniki wagowe mają za zadanie różnicować stosunkowo mniejszy wpływ entropii małolicznych skupień w stosunku do tych dużych. Ze względu na bardzo nierównomierny rozkład decyzji w skupieniach analizowanych w ramach rozprawy doktorskiej, miara entropii byłaby nieprzydatna, a jej wyniki – nieużyteczne w kontekście oceny jakości generowanych skupień.

Czystość (ang. *purity*) to podobnie jak entropia miara określająca jak dużo miejsca w danej grupie zajmują obiekty należące do jednej klasy:

$$p_i = \max_{i=1,\ldots,K} p_{ij}.$$

Jak widać, wskaźnik ten to maksymalne prawdopodobieństwo napotkania na obiekt $j-$tej klasy w $i-$tym skupieniu.

Całkowita czystość wyraża się jako:

$$p = \sum_{i=1}^{K} \frac{m_i}{N} \cdot p_i$$

i jest interpretowana analogicznie do miary całkowitej entropii.

Dokładność (ang. *precision*) w systemach wyszukiwania informacji – jest stosunkiem liczby obiektów relewantnych wyszukanych do wszystkich znalezionych. W odniesieniu do procesu oceny jakości grupowania możemy stwierdzić, że jest stosunkiem liczby obiektów należących do jednej klasy i będących w jednej grupie do liczby wszystkich obiektów należących do tej klasy. Jej wartość liczbowa obliczana jest analogicznie jak w przypadku miary p_{ij} omawianej w kontekście entropii.

Kompletność (ang. *recall*) podobnie jak dokładność, miary tej używa się również w systemach wyszukiwania informacji. Jest to stosunek liczby obiektów relewantnych wyszukanych do liczby wszystkich relewantnych w systemie. W kontekście oceny jakości grupowania jest to stosunek liczby obiektów należących do jednego skupienia i jednej klasy m_{ij} do liczby wszystkich dokumentów etykietowanych daną klasą m_j:

$$RECALL_{ij} = \frac{m_{ij}}{m_j}.$$

F-miara (ang. *F-measure*) jest definiowana jako kombinacja kompletności i dokładności wraz ze współczynnikiem wagowymi β:

$$F_{ij} = \frac{(1+\beta) \cdot p_{ij} \cdot RECALL_{ij}}{\beta \cdot p_{ij} + RECALL_{ij}}.$$

Można zauważyć, że jest to ważona średnia harmoniczna wartości kompletności i dokładności ze współczynnikiem β, którego celem jest zmiana proporcji średniej ważonej pomiędzy kompletnością a dokładnością. Klasyczna F-miara przyjmuje współczynnik β równy 1.

	Zgodna klasa	Niezgodna klasa
Zgodna grupa	a	b
Niezgodna grupa	c	d

Tabela 6.1: Tabela współczynników miar zorientowanych na podobieństwo

Miary zorientowane na podobieństwo

Wyżej przedstawione miary brały pod uwagę zgodność klasyfikacji do grup zgodnie ze znanymi wcześniej klasami. Druga grupa metod oceny zewnętrznej sprawdza podobieństwo struktur użytych w czasie grupowania (np. macierzy podobieństwa) z idealnymi strukturami znanymi lub wyznaczonymi wcześniej.

W niektórych miarach do bardziej przejrzystego zaprezentowania wzorów i definicji można posłużyć się tabelą 6.1 wzorowaną na miarach znanych z teorii błędu [138].

Tabelę 6.1 należy rozumieć w taki sposób, iż po wykonaniu procesu grupowania współczynniki a, b, c, d określają konkretne wartości liczbowe zgodne z etykietami. Przykładowo, współczynnik a to liczba obiektów pogrupowanych do jednego skupienia zgodnego z wcześniej zdefiniowaną optymalną klasą.

Najważniejsze miary zorientowane na podobieństwo przedstawione są poniżej:

Statystyka Randa Współczynnik Randa określa stosunek wszystkich "dobrych przyporządkowań" do liczby wszystkich przyporządkowań. Przez "dobre przyporządkowanie" należy rozumieć sklasyfikowanie dwóch obiektów do tej samej grupy jeśli tylko znajdują się one w tej samej klasie lub przyporządkowanie do różnych grup jeśli znajdują się one w różnych klasach. Współczynnik przyjmuje wartości z przedziału $[0 \ldots 1]$. Wartość bliska 0 oznacza, że schemat grupowania zupełnie nie pokrywa się z klasami, natomiast $RAND$ równy w przybliżeniu 1 jest oznaką bardzo dużej zgodności schematu grupowania oraz schematu klas:

$$RAND = \frac{d + a}{a + b + c + d}.$$

Miara Jaccarda Miara Jaccarda jest współczynnikiem podobnym do statystyki Randa z drobną różnicą polegającą na niebraniu pod uwagę

obiektów z różnych klas przyporządkowanych do różnych grup. Takie podejście zwiększa wpływ czynnika określającego zgodność przyporządkowania:

$$JACCARD = \frac{a}{a+b+c}.$$

Γ **Statystyka Huberta** Jest to kolejny współczynnik operujący na wartościach porównujących klasę i grupę. Największy wpływ na jego wartość mają całkowicie poprawne przyporządkowania (zgodna grupa i zgodna klasa oraz niezgodna grupa i niezgodna klasa) [169]:

$$\Gamma = \frac{(a+b+c+d)\cdot a - (a+b)\cdot(a+c)}{\sqrt{(a+b)\cdot(a+c)\cdot(c+d)\cdot(b+d)}}.$$

6.2 Ocena jakości procesu wnioskowania w systemach ekspertowych

Jakość systemów ekspertowych będzie uzależniona od jakości poszczególnych elementów wchodzących w skład takiego systemu. W poprzednich rozdziałach omówiona była rola niesprzeczności bazy wiedzy, co niewątpliwie wpływa na wzrost szybkości przeszukiwania bazy wiedzy. W celu dalszego przyspieszenia – powstały omówione wcześniej algorytmy przeszukiwania bazy wiedzy (m.in. RETE, LEAPS, itp.). Szybkie odnalezienie odpowiedniej reguły jest jednym z mierzonych czynników poprawiających efektywność SWD. Drugim czynnikiem badanym przez autora jest zdolność do uaktywniania reguł w warunkach niepewności. W przypadku małej liczności zbioru faktów, klasyczne systemy wspomagania decyzji nie uaktywnią żadnej z reguł, a co za tym idzie – nie dostarczą żadnej nowej wiedzy z systemu.

Proponowane podejście stara się rozwiązać te dwa problemy poprzez kontrolowane uaktywnianie reguł o odpowiednim stopniu pewności. Dzięki temu zadanie eksploracji nowej wiedzy, nawet przy małej liczbie faktów, może być realizowane.

6.2.1 Przegląd dostępnych rozwiązań

Jedną z najwcześniejszych prób ewaluacji systemów ekspertowych można odnaleźć w pozycji [130]. Autorzy proponują tam całościowe podejście do

problemu i ocenę nie tylko narzędzia informatycznego, ale całego procesu tworzenia SWD. Ich działanie opiera się na ocenie pod względem następujących kryteriów:

- charakterystyki zastosowania: ocena problemu, dziedziny i wykonalności projektu,
- możliwości narzędzia,
- metryki, czyli miar oceny możliwości wykorzystywanych narzędzi do budowy systemu ekspertowego,
- technik oceny, czyli sposobu zastosowania powyższych metryk,
- kontekstu.

W pracy podane są szczegółowe propozycje oceny poszczególnych kryteriów oraz przykład oceny rzeczywistych systemów ekspertowych.

Autor publikacji [147] zwraca uwagę na powszechny "syndrom 95%". Polega on na tym, iż zwykle prototyp systemu ekspertowego oceniany jest na podstawie procentowej miary pomyślnych odpowiedzi. Nierzadko jest ona bardzo wysoka (tytułowe "95%"), co bywa mylące ze względu na odpowiedni dobór przypadków testowych. Problem ten jest analogiczny do zjawiska przeuczania klasyfikatorów. Autor przedstawia ocenę dwóch prototypowych systemów ekspertowych za pomocą kilku kryteriów takich jak zdolność do aktualizacji, łatwość użytkowania czy czas potrzebny do poprawnego zaprojektowania systemu ekspertowego. Niestety, większość z tych kryteriów jest subiektywna i ciężka do zastosowania praktycznego. Autor konkluduje swój wywód smutnym wnioskiem powstawania dużej liczby prototypów systemów ekspertowych, które jednakże nie zostają pomyślnie wdrożone u użytkownika końcowego.

Grupa autorów z Kanady [47] proponuje zastosowanie metod znanych z inżynierii oprogramowania do oceny systemów ekspertowych. Początkowo, niezbędne jest wykonanie *specyfikacji* działania systemu, która pozwoli na odkrycie realnych potrzeb użytkownika. Kolejnym krokiem jest *weryfikacja* wewnętrznych sprzeczności w bazie wiedzy oraz końcowa *walidacja* zgodności z wcześniej zdefiniowaną specyfikacją. Autorzy proponują częściową automatyzację weryfikacji bazy wiedzy za pomocą swojego narzędzia o nazwie "COVER".

Wyczerpujące i całościowe kompendium wiedzy nt. weryfikacji, walidacji i ewaluacji systemów ekspertowych przedstawia pozycja [166]. Jest to

pozycja pisana przez praktyków jako podręcznik oceniania systemów ekspertowych. Pomocne i praktyczne rady oraz metody oceny każdego aspektu i stadium życia systemu poparte są tu praktycznymi rozwiązaniami i wskazówkami.

Autorzy stosunkowo nowych opracowań [95] i [134] proponują użycie ankiet w celu weryfikacji systemów ekspertowych. Pierwsza pozycja rozpoczyna się wyczerpującym przeglądem literatury dziedzinowej aby w kolejnych rozdziałach podać gotowe przykłady ankiet ewaluacyjnych wraz z wagą punktową każdej odpowiedzi i progami punktowymi oceniającymi systemy ekspertowe. Druga z pozycji pokazuje inny (znacznie uproszczony) wariant ankiety ewaluacyjnej.

Jak widać, zdecydowana większość autorów proponuje subiektywną ewaluację systemów ekspertowych. W tej rozprawie, prócz takowych, zastosowane zostaną również inne, obiektywne i automatyczne metody służące do wykazania przydatności i poprawności prezentowanego rozwiązania.

6.2.2 Efektywność przeszukiwania bazy wiedzy

Efektywność przeszukiwania bazy wiedzy jest zwykle określana poprzez dwa parametry: kompletność oraz dokładność wyszukiwania. Pierwszy z nich określa stopień przeszukania bazy wiedzy i odnalezienia wszystkich reguł relewantnych (czyli jak "pełne" jest wyszukiwanie reguł), drugi mówi jak duża część odnalezionych reguł jest adekwatna w stosunku do aktualnie poszukiwanych. Omówione tutaj miary są analogiczne do tych używanych przy ocenie zewnętrznej zorientowanej na klasyfikację przedstawionych w rozdziale 6.1.2. W przypadku, gdy przeszukiwanie skupień reguł zakończy się wyborem reguły (skupienia), która jest nie w pełni pokryta zbiorem faktów, przedstawione tu miary będą odpowiednio zmniejszone.

Aby łatwiej zrozumieć przedstawione pojęcia, wprowadźmy oznaczenia jak w tabeli 6.2:

	Skupienia (reguły) relewantne	Skupienia (reguły) nierelewantne
Skupienia (reguły) wyszukane	a	b
Skupienia (reguły) niewyszukane	c	d

Tabela 6.2: Tabela współczynników miar opartych na teorii błędu

Do pomiaru efektywności przeszukiwania bazy wiedzy można użyć następujących miar:

Kompletność – (ang. *recall* [76]) to iloraz liczby skupień (reguł) relewantnych wyszukanych i sumy liczby skupień (reguł) relewantnych (wyszukanych oraz niewyszukanych):

$$K = \frac{a}{a+c}.$$

Definiowana jest również jako zdolność systemu do wyszukania wszystkich reguł relewantnych.

Dokładność [76] – (ang. *precision*) to iloraz liczby skupień (reguł) relewantnych wyszukanych i sumy liczby wszystkich skupień (reguł) wyszukanych (relewantnych i nierelewantnych):

$$D = \frac{a}{a+b}.$$

Optymalizacja parametrów systemów wspomagania decyzji będzie głównie dotyczyć parametru dokładności, nawet kosztem kompletności. W procesie wyszukiwania reguł ważniejszym jest odnalezienie reguł dokładnych, tak aby były one możliwe do uaktywnienia i można było dzięki temu uzyskać nową wiedzę z systemu. Mniejsze znaczenie będzie miało odnalezienie wszystkich możliwych reguł.

Procedura wyznaczania stopnia niepełności konkluzji uaktywnianych reguł (lub niepełności całego odnalezionego skupienia wybranego do procesu wnioskowania) została opisana w podrozdziale 5.3.5.

6.2.3 Efektywność wnioskowania

W rozdziale 3 przedstawione zostały algorytmy wnioskowania w SWD. Ich działanie oparte jest na bazie wiedzy tworzonej przez inżyniera wiedzy. Od jakości tej wiedzy zależy bezpośrednio efektywność wnioskowania, albowiem żaden z algorytmów wnioskowania nie będzie w stanie doprowadzić do poprawnych i użytecznych wniosków bez użycia poprawnych danych. Tworzenie i organizacja wiedzy w bazie wiedzy jest często niedocenianą częścią optymalizacji procesów wnioskowania. Problem niepełności wiedzy dodatkowo komplikuje tą sytuację. Często bezcelowe wyszukiwanie reguły, której nie ma w systemie również prowadzi do spadku efektywności wnioskowania.

W klasycznym podejściu efektywność wyszukiwania reguł (ang. *conflict set resolution*) bardzo silnie zależeć będzie od kolejności zapisu reguł w bazie wiedzy. Płaska struktura reguł pozwala na wyszukiwanie reguł relewantnych w czasie $O(n)$, co dla dużych systemów jest dość długim procesem. Należy się zastanowić w jakiej kolejności w takim razie zapisywać reguły w bazie wiedzy, aby wyszukiwanie ich było najszybsze. Precyzyjna odpowiedź niestety jest niemożliwa, albowiem zależy to od konkretnego wnioskowania uruchamianego przez użytkownika. W idealnym przypadku, kolejność zapisu reguł w bazie wiedzy nie powinna mieć wpływu na czas wnioskowania. Modyfikacja polegająca na zapisie hierarchicznym bazy wiedzy nie tylko przyspiesza wyszukiwanie dzięki wyszukiwaniu przy użyciu drzewa binarnego reguł, ale również (przy założeniu korzystania z algorytmów deterministycznych) uwalnia system od zależności związanej z kolejnością dopisywanych reguł.

Omówione wcześniej strategie doboru reguł (ang. *conflict resolution methods*) w pewnym stopniu wpływają na jakość generowanej wiedzy. Przy założeniu niesprzeczności wiedzy w bazie wiedzy, ich wpływ nie powinien zostać przeceniany.

Zaproponowana struktura hierarchiczna posiada liczne zalety omówione wcześniej, a dyskutowane w szczególności w kolejnych rozdziałach. Niektóre z nich to przyspieszenie procesu wyszukiwania reguł relewantnych, możliwość wyszukiwania całych skupień reguł najbardziej adekwatnych w stosunku do aktualnego zbioru faktów i możliwość odnajdywania i uaktywniania reguł, których nie wszystkie przesłanki są spełnione. Kolejną zaletą jest możliwość uogólniania wiedzy. Na każdym poziomie hierarchii skupień reguł (budowanej przez algorytm AHC) możliwym jest wyznaczenie reprezentantów skupień skutecznie odzwierciedlających wiedzę na wyższym poziomie abstrakcji.

6.2.4 Walidacja grupowania

Walidacja grupowania (lub ewaluacja grupowania – ang. *cluster evaluation / validation*) jest procesem wspomagającym grupowanie. Ma on na celu ocenę bieżącą grupowania, dzięki której można dostrajać parametry wybranego algorytmu w celu uzyskania najlepszych wyników. Do zadań walidacji należą między innymi [150]:

1. porównanie podziałów końcowych w celu wyłonienia najlepszego,

2. ocena empiryczna (na podstawie znanych danych) grupowania; mając dany optymalny podział danych (lub eksperta, który taki podział może wyznaczyć) można porównać wygenerowany podział na grupy i ocenić jego stopień zbliżenia do podziału optymalnego,

3. dla algorytmów, które tego wymagają, dobór odpowiednich parametrów, takich jak liczba grup, optymalna miara odległości, liczba skupień, gęstość siatki gridowej, itp.,

4. podjęcie decyzji, czy zaprezentowane dane przejawiają tendencję do grupowania (tj. czy występują w nich naturalne grupy możliwe do odkrycia),

5. ocena użyteczności grupowania; fakt znalezienia formalnie najbardziej poprawnego podziału nie zawsze gwarantuje, że podział ten jest użyteczny ze względu na potrzeby użytkownika.

6.3 Złożoność obliczeniowa rozwiązania

Jedną z uniwersalnych metod oceny rozwiązania informatycznego jest oszacowanie jego złożoności obliczeniowej [27].

W przypadku przedstawianego systemu istotnymi będą złożoność pamięciowa oraz czasowa. Pierwsza z nich określa szybkość przyrostu ilości pamięci w stosunku do zwiększającego się rozmiaru danych wejściowych, druga z nich – szybkość przyrostu czasu obliczeń.

W obrębie SWD jedną z bardziej czasochłonnych operacji jest proces wnioskowania na który składa się proces wyszukiwania reguły oraz jej uaktywnienia. Wnioskowanie progresywne wymaga wyszukiwania wszystkich reguł możliwych do uaktywnienia i zwykle jest przeprowadzane w czasie liniowym. Proces ten powinien być powtarzany aż do momentu uaktywnienia wszystkich reguł w systemie. Wyszukiwanie sterowane celem z kolei jest procesem rekurencyjnym i wywoływanym wielokrotnie aż do momentu udowodnienia zakładanej hipotezy.

Problem wiedzy niepełnej nie będzie miał tutaj wielkiego wpływu na szacowanie złożoności obliczeniowej. Zarówno wyszukiwanie jak i uaktywnianie reguł wygląda w sposób identyczny w obu przypadkach. Jedyną różnicą jest rezultat, który w przypadku wnioskowania z użyciem wiedzy niepełnej częściej może być negatywny.

6.3.1 Złożoność obliczeniowa algorytmów grupowania

Algorytmy grupowania charakteryzują się stosunkowo dużą złożonością obliczeniową (na poziomie wielomianowym drugiego i trzeciego stopnia). Algorytm AHC oraz $mAHC$ użyte w tej rozprawie należą do kategorii złożoności czasowej $O(n^2)$. Należy zauważyć jednak, że grupowanie konieczne jest do wykonania jedynie raz, a jego rezultaty mogą być zapamiętane w pamięci nieulotnej.

Przywołane w rozprawie algorytmy niehierarchiczne (m.in. k-means), postępują w zgoła odmienny sposób od algorytmów hierarchicznych. W pierwszym kroku algorytmu konieczne jest wykonanie dla każdego z n obiektów k operacji obliczenia odległości do środków każdej z k grup. W sumie daje to złożoność na poziomie $O(n \cdot k)$. W każdym z następnych t kroków (prób poprawy i utworzenia lepszych jakościowo grup) obliczane jest n odległości obiektów do wybranych centroidów, a po dalszych k operacjach wyznaczane są nowe centroidy. Stąd, złożoność obliczeniowa tej części to $O(t \cdot (n + k))$. Reasumując, złożoność obliczeniowa algorytmu k-means kształtuje się na poziomie:

$$O(n \cdot k) + O(t \cdot (n + k)).$$

W czasie grupowania dążymy do zmniejszenia rozmiaru danych, stąd $k << n$. Liczba iteracji t też jest zwykle znacznie mniejsza od liczby próbek danych n, stąd ostatecznie złożoność obliczeniowa algorytmu k-means wynosi $O(n)$.

Niestety, jak zostało to wykazane w rozdziale 5 algorytmy niehierarchiczne mają szereg wad, z których najpoważniejszą jest niedeterminizm. Algorytmy hierarchiczne z kolei niwelują te wady za cenę wyższej złożoności obliczeniowej. Algorytmy aglomeracyjne w pierwszym kroku budują macierz podobieństwa o rozmiarze $n \times n$. Macierz ta jest wypełniana tylko w połowie ze względu na jej symetrię wzdłuż przekątnej. Do wypełnienia potrzeba jednak $n + (n - 1) + (n - 2) + \cdots + 1 = \frac{n \cdot (n+1)}{2}$ kroków, co daje złożoność rzędu $O(n^2)$. W kolejnych $n - 1$ krokach należy przeszukać tą strukturę w celu odnalezienia minimalnej wartości, co również, w przypadku korzystania z przeszukiwania liniowego, ma miejsce ze złożonością rzędu $O(n^2)$. W każdym kroku należy również zaktualizować wartość n odległości ze względu na połączenie obiektów w grupy. Fakt, iż całkowita liczba obiektów maleje z każdym krokiem grupowania o 1 (innymi słowy: stopień macierzy podobieństwa zmniejsza się o 1 w każdym kroku grupowania) nie ma wpływu na obliczenia związane ze złożonością obliczeniową.

Reasumując, algorytmy hierarchiczne charakteryzują się złożonością obliczeniową na poziomie:

$$O(n^2) + O(n-1) \cdot O(n^2) = O(n^3).$$

Algorytm *AHC* jest przykładem algorytmu hierarchicznego aglomeracyjnego i jako taki ma złożoność obliczeniową ustaloną na poziomie $O(n^3)$. Najkosztowniejszą operacją jest wyszukiwanie minimalnej (lub maksymalnej) wartości podobieństwa w macierzy podobieństwa. Rozważaną modyfikacją jest skorzystanie z bardziej wyrafinowanych rozwiązań tego problemu co z pewnością skutkować będzie zmniejszeniem czasu obliczeń.

Algorytm *mAHC* w wersji znanej z literatury charakteryzuje się zmniejszoną liczbę iteracji ze względu na określone kryterium stopu kończące pracę po utworzeniu k grup. Stąd też złożoność obliczeniowa algorytmu kształtuje się na poziomie $O(k \cdot n^2)$. Ze względu na fakt, że zwykle $k << n$, złożoność obliczeniowa algorytmu dla małych wartości k zmniejsza się do wartości $O(n^2)$. Wystarczy jednak sytuacja, w której $k \approx n$ (liczba iteracji algorytmu będzie w przybliżeniu równa liczbie obiektów w systemie, algorytm będzie zachowywał się podobnie jak *AHC*) i złożoność obliczeniowa wzrośnie do $O(n^3)$.

Przedstawiony w rozprawie algorytm *mAHC* buduje pełną hierarchię skupień, a następnie pozwala na dynamiczne przycinanie otrzymanego dendrogramu. Warunkuje to niestety złożoność obliczeniową na poziomie zwykłego algorytmu *AHC* równą $O(n^3)$. Przycinanie drzewa jest operacją o niskiej złożoności obliczeniowej i nie ma wpływu na całkowitą złożoność algorytmu.

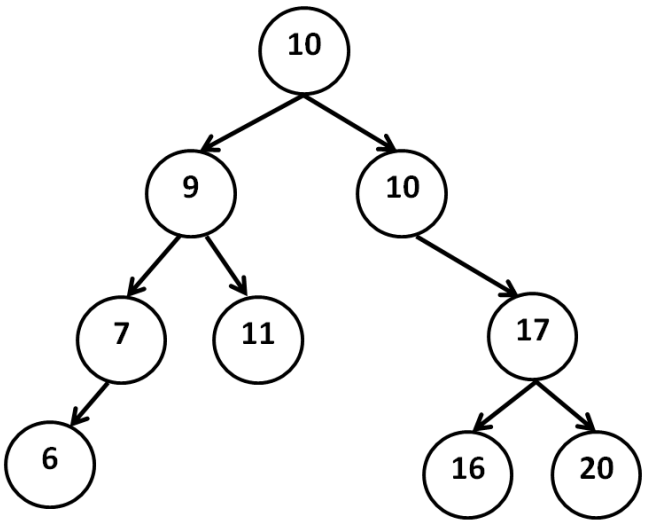
6.3.2 Złożoność obliczeniowa wnioskowania

Wnioskowanie jest procesem nietrywialnym i trudnym do oszacowania pod względem złożoności obliczeniowej. Zarówno budowa bazy wiedzy, kolejność zapisu reguł czy strategia uaktywniania reguł mogą w znaczący sposób zmieniać wartość czasu potrzebną do przeprowadzenia procesu wnioskowania. Należy również pamiętać, że (zwłaszcza w kontekście wiedzy niepełnej) wnioskowanie może zwrócić wynik negatywny. W takim przypadku użytkownik końcowy zwykle dokonuje weryfikacji danych wejściowych i przeprowadza algorytm ponownie. Wnioskowanie w rmach klasycznej bazy wiedzy (bez grup reguł) bardzo często bywa nieefektywne.

W przypadku optymistycznym cel wnioskowania znajdzie się na liście znanych faktów i wnioskowanie zakończone zostanie po przeglądnięciu listy

m faktów. Lista faktów jest zwykle dość krótka, więc można przyjąć, że dzieje się to w stałym czasie.

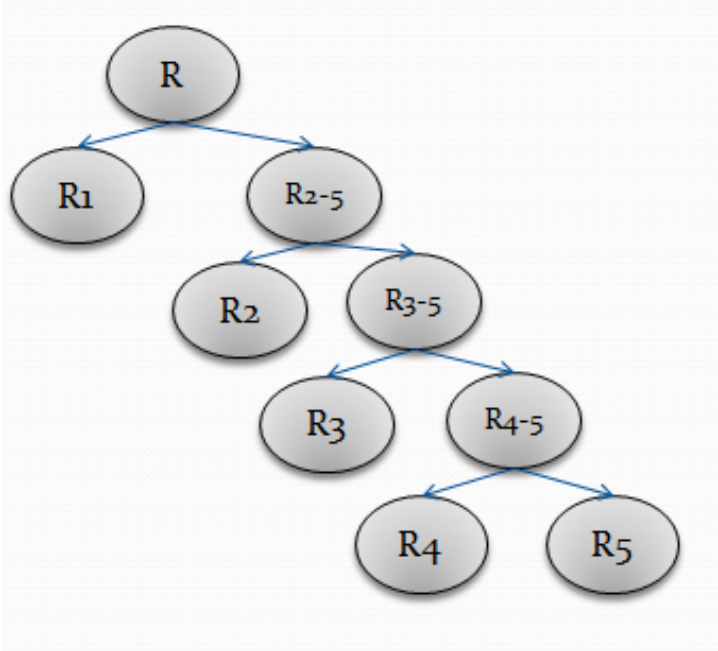
W pesymistycznym przypadku konieczne będzie przejrzenie n reguł, z których każda posiada do z przesłanek, które należy skonfrontować z m faktami. W zastosowaniach praktycznych na szczęście zarówno z jak i m są znacznie mniejsze od liczby reguł n ($z, m << n$), co w konkluzji daje złożoność obliczeniową na poziomie $O(n)$. Należy tu również zauważyć, że bazę wiedzy można przeglądać wielokrotnie. W szczególności, przy najgorszej możliwej permutacji kolejności reguł w bazie wiedzy, aby uaktywnić pierwszą regułę należy przejrzeć n reguł. Wygenerowana konkluzja dołączy do bazy wiedzy co pozwoli na uaktywnienie $n-1$ reguły w bazie wiedzy, itd. Reasumując, wygenerowanych zostanie n nowych faktów za każdym razem przeglądając $n-i$ reguł, gdzie i to kolejna iteracja algorytmu wnioskującego. Łatwo zauważyć, że złożoność takiego procesu dąży nawet do wartości $O(n^3)$.



Rysunek 6.4: Drzewo binarne

Proponowana w rozprawie struktura skupień reguł pozwala na bardzo szybkie wyszukiwanie o logarytmicznej złożoności obliczeniowej $O(log_2 n)$ [27]. Za pomocą algorytmów analizy skupień płaska struktura bazy wiedzy przekształcona zostaje do postaci drzewa binarnego (rys. 6.4). Należy tu jednak zauważyć, że tak określona złożoność średnia nie uwzględnia faktu tzw. braku zbilansowania drzewa. Jeśli drzewo będzie zbudowane optymalnie, to znaczy tak, aby liczba węzłów na każdym poziomie była identyczna,

wtedy istotnie, pesymistyczna złożoność obliczeniowa wyszukiwania dowolnego elementu będzie $\Theta(log_2 n)$. Jednakże w miarę rozrostu jednej z gałęzi, pesymistycznie do długości n (czyli zamiast drzewa mamy do czynienia z liniową ciągiem reguł) złożoność ta zwiększa się do wartości $\Theta(n)$. Omawiana sytuacja przedstawiona jest na rys. 6.5.



Rysunek 6.5: Niezbilansowane (nieoptymalne) drzewo binarne

Istniejące rozwiązania pozwalające na bilansowanie drzew (np. drzewa czerwono-czarne [27]) nie nadają się niestety do zastosowania w problemie budowania hierarchii skupień reguł. Jedynymi metodami pozwalającymi na optymalizację dendrogramu jest optymalizacja parametrów algorytmu grupującego co jest kluczowym elementem tej rozprawy doktorskiej.

6.4 Podsumowanie

Jak przedstawiono powyżej, problem optymalizacji efektywności systemu ekspertowego jest pojęciem nietrywialnym. Co więcej, mamy tu do czynienia z optymalizacją wielokryterialną. Jeśli dodamy do tego niepełność wiedzy, problem ten staje się jeszcze trudniejszy.

Przedstawiane rozwiązanie próbuje radzić sobie z optymalizacją wnioskowania w warunkach wiedzy niepełnej poprzez użycie hierarchicznej struk-

tury grup reguł zbudowanych dzięki algorytmowi analizy skupień AHC. Jak zostanie to wykazane, dzięki skorzystaniu z tej hierarchicznej struktury bazy wiedzy możliwa jest optymalizacja:

1. bazy wiedzy poprzez utworzenie struktury hierarchicznej reguł, dzięki czemu ich wyszukiwanie będzie znacznie przyspieszone. Niestety, kosztem jest konieczność jednokrotnego stworzenia takiej hierarchicznej struktury bazy wiedzy. Złożoność obliczeniowa tego kroku wynosi $O(n^2)$,

2. modułu wnioskowania poprzez:

   - szybkie wyszukiwanie reguł korzystając ze struktury hierarchicznej. Odnajdywanie reguł w takiej strukturze bazy jest klasycznym przeszukiwaniem drzewa binarnego, stąd złożoność obliczeniowa tego procesu to $O(log_2 n)$.
   - uaktywnianie także reguł nie pokrytych całkowicie zbiorem faktów, co pozwala eksplorować więcej wiedzy niż na to pozwalałoby klasyczne wnioskowanie. W wyniku wnioskowania będzie można przedstawić użytkownikowi dodatkowe informacje o elementach koniecznych do udowodnienia przed przeprowadzeniem wnioskowania dokładnego. Jeśli to nie będzie możliwe, zostanie zaproponowane wnioskowanie oparte na autorskiej metodzie współczynników IF omówionych w rozdziale 6.3.

W celu oceny jakości przedstawianego systemu w części eksperymentalnej przedstawione zostaną wyniki badań następujących parametrów:

1. Kompletność (rozumiana, jako stosunek liczby reguł relewantnych odnalezionych do liczby wszystkich reguł relewantnych w systemie).

2. Dokładność (rozumiana jako stosunek liczby reguł relewantnych odnalezionych do liczby wszystkich reguł odnalezionych przez system).

3. Liczbę reguł w wygenerowanym skupieniu.

4. Zysk czasowy przeglądu bazy i odnajdywania reguł relewantnych z wykorzystaniem skupień wygenerowanych algorytmem AHC lub mAHC w stosunku do przeglądu klasycznymi metodami.

5. Stosunek liczby przypadków pomyślnego przeprowadzenia wnioskowania w warunkach wiedzy niepełnej do wszystkich prób wnioskowania.

W kontekście analizy złożoności obliczeniowej przedstawianego rozwiązania, można również powiedzieć, że:

1. Część grupująca przedstawianego rozwiązania cechuje się złożonością obliczeniową na poziomie $O(n^3)$.

2. Część wyszukująca reguły możliwe do uaktywnienia (także w kontekście wiedzy niepełnej) ma złożoność obliczeniową na poziomie $O(log_2 n)$.

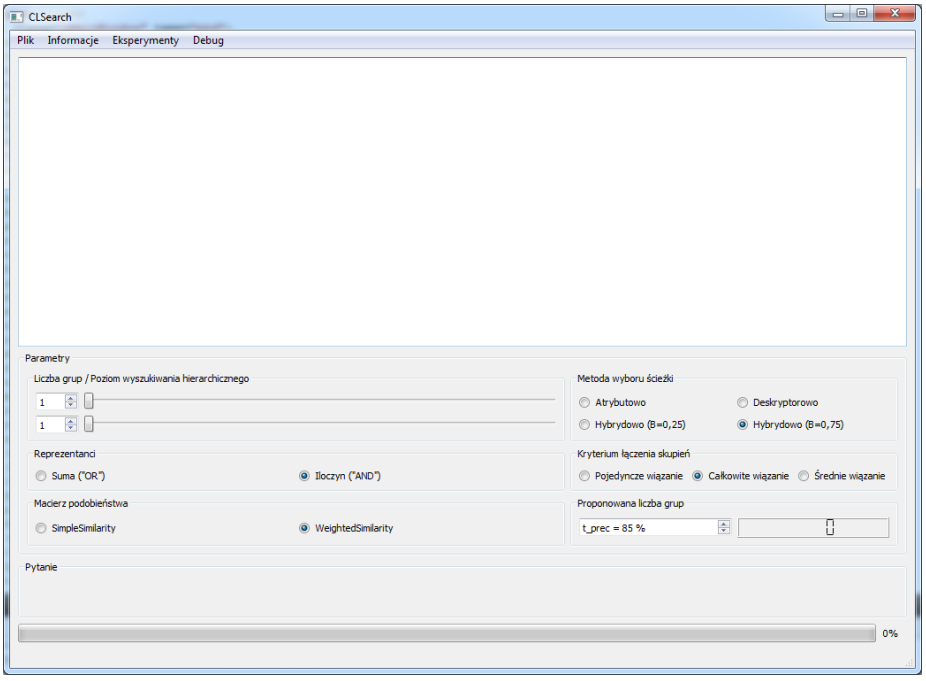
3. Część uaktywniająca wybrane reguły o wartościach współczynnika IF większego od zadanego progu ma złożoność obliczeniową szacowaną na $O(r)$, gdzie r to liczba reguł wewnątrz odnalezionego skupienia.

Złożoność pamięciowa algorytmu jest również stosunkowo duża, głównie ze względu na konieczność pamiętania całej macierzy rozróżnialności o rozmiarze $n \cdot n$. Analizowana struktura hierarchiczna zajmuje już tylko $2n - 1$ komórek pamięci.

Rozdział 7

Projekt systemu

W celu zaprezentowania przedstawionej metody w praktyce, autor dokonał implementacji rozważanych algorytmów i podejść. W tym celu powstał system *CLSearch*, którego główne okno widoczne jest na rys. 7.1.



Rysunek 7.1: Główne okno autorskiego programu CLSearch.

Autor dokonał implementacji przy użyciu środowiska Qt w wersji 5.1 [32]. Do prawidłowego działania system nie wymaga żadnych dodatkowych

bibliotek. Ze względu na użycie dużych struktur danych wymagany jest 64bitowy system operacyjny*.

7.1 Dokumentacja systemu

Program był testowany w środowisku *Windows XP, Windows 7, Linux* w wersjach 32bitowych oraz 64bitowej. Wersja 32bitowa nadaje się do przeprowadzania wnioskowania w mniejszych bazach wiedzy. Minimalne wymagania systemowe przedstawione są w tabeli 7.1.

Procesor o częstotliwości taktowania 1,6 GHz
2GB pamięci RAM
50 MB wolnego obszaru dysku twardego
Monitor o rozdzielczości 1366x768 px
Napęd CD-ROM
Klawiatura, mysz

Tabela 7.1: Minimalne wymagania sprzętowe aplikacji CLSearch.

W celu zapewnienia optymalnego działania, wielkość pamięci RAM powinna wynosić co najmniej 8GB. W przeciwnym wypadku aplikacja wykorzystywać będzie plik stronicowania, co znacząco zmniejszy jej wydajność. Należy jednakże zauważyć, że tak duże wymagania ma jedynie część grupująca. Użycie wnioskowania w wygenerowanej wcześniej strukturze może zostać przeprowadzone na znacznie słabszej konfiguracji sprzętowej.

7.1.1 Instrukcja użytkownika

Aby skorzystać z systemu należy przystąpić do jego instalacji. Ta sprowadza się do przekopiowania folderu dołączonego na płycie CD na dysk twardy komputera i uruchomienia pliku *CLSearch.exe*. Na płycie znajdują się dwie wersje oprogramowania: 32bitowa oraz 64bitowa. Wybór podyktowany jest bitowością systemu operacyjnego na którym uruchamiany jest system.

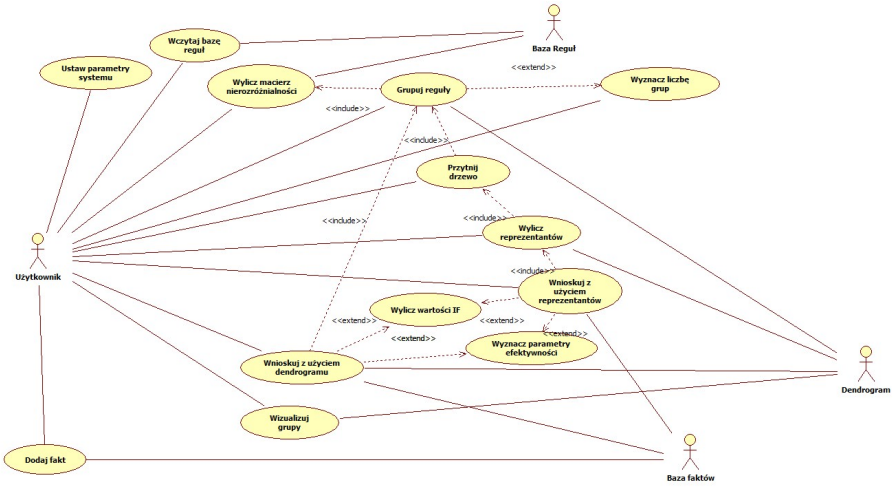
Aplikacje stworzone przy użyciu Qt charakteryzują się wysoką kompatybilnością międzyplatformową. W celu uzyskania wersji dla komputerów Linux wystarczy dokonać rekompilacji kodu aplikacji wybierając odpowiednią platformę sprzętowo-programową. Odbywa się to bez zmiany kodu aplikacji.

Po uruchomieniu aplikacji, użytkownikowi prezentowany jest ekran startowy wraz z komunikatem *Gotowy do pracy.*

*Potrafiący zaadresować więcej niż 4GB pamięci operacyjnej.

7.1.2 Diagram przypadków użycia

Na rys. 7.2 prezentowane są funkcje systemu wraz z informacjami o ich wzajemnych zależnościach.



Rysunek 7.2: Diagram przypadków użycia dla systemu

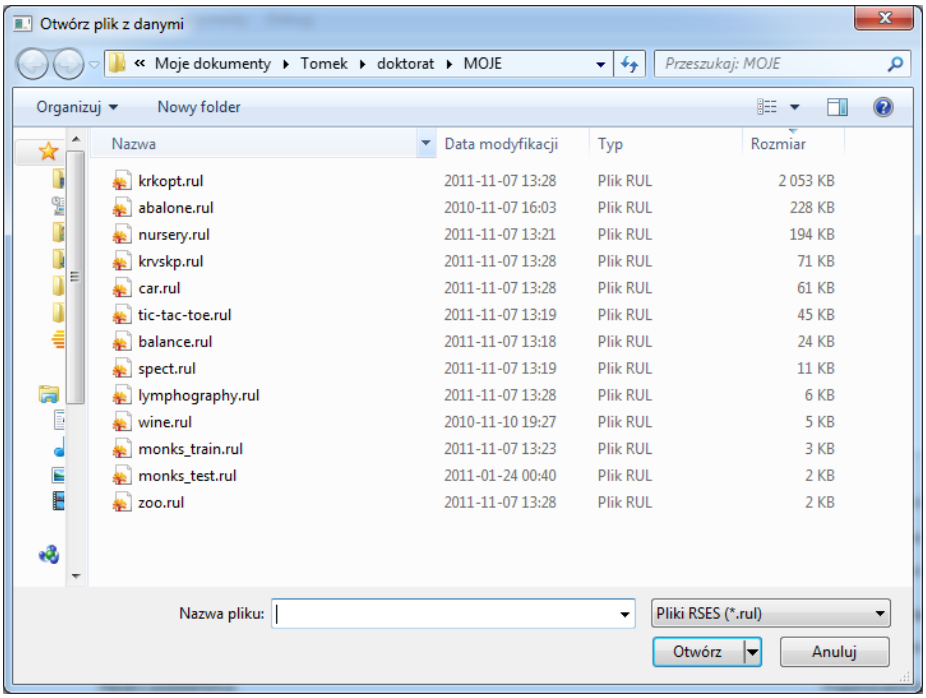
Na diagramie wyróżnić można czterech aktorów: użytkownika systemu, bazę reguł, dendrogram (czyli bazę reguł daną w postaci hierarchicznej) oraz bazę faktów. Poszczególne funkcjonalności systemu modyfikują lub działają z użyciem tych wymienionych aktorów. Należy zauważyć, że niezależnie od sposobu przeprowadzenia wnioskowania, użytkownik może wyliczyć parametry efektywności systemu oraz wartości współczynników IF. Jednocześnie niemożliwym jest przeprowadzenie wnioskowania przy użyciu reprezentantów bez ich uprzedniego wyznaczenia (co jest niemożliwe bez przycięcia drzewa, itd.). Przypadek użycia *Wyznacz liczbę grup* może być wykonywany niezależnie w czasie grupowania.

7.2 Budowa hierarchicznej bazy wiedzy

W celu rozpoczęcia budowy hierarchicznej bazy wiedzy należy wczytać bazę wiedzy w postaci pliku formatu RSES [13] co prezentuje rys. 7.3. Plik ten zawiera reguły zapisane w kolejności ich wygenerowania. Jest to płaska struktura regułowej bazy wiedzy.

Źródłowy format bazy wiedzy jest przedstawiony na rys. 7.4.

Baza wiedzy w formacie RSES zapisywana jest w następujący sposób:



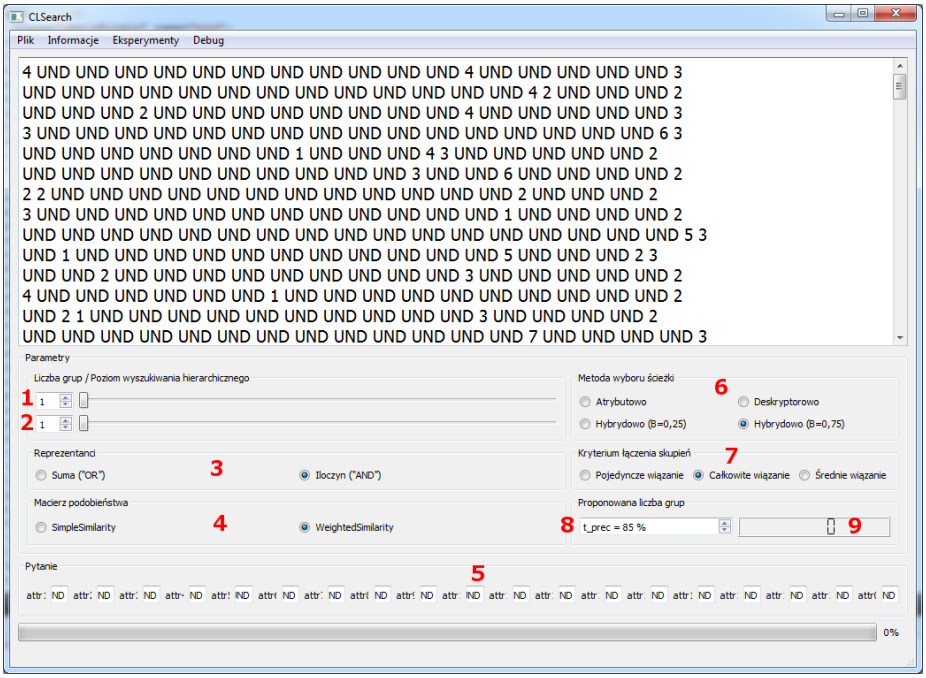
Rysunek 7.3: Wczytywanie bazy danych

- Pierwsza linijka to zawsze nazwa wczytywanego zbioru reguł.
- Druga linijka określa liczbę atrybutów warunkowych z i decyzyjnych m.
- Kolejne $z + m$ linijek to nazwy i typy atrybutów warunkowych i decyzyjnych. Liczba na końcu linijki opisującej atrybuty warunkowe określa długość maksymalnej wartości atrybutu liczoną w znakach drukowanych.
- Kolejna linijka określa liczbę u wartości atrybutu decyzyjnego. Tradycyjnie atrybut decyzyjny wpisywany jest jako ostatni w kolejności występowania w regule, jednakże nie jest to wymagane.
- Następne u linii określa wartości atrybutu decyzyjnego.
- W ostatniej sekcji pliku po nagłówku określającym liczbę n reguł występujących w bazie wiedzy następuje n linii z regułami systemu ekspertowego. Ostatnia wartość w każdej linii określa tzw. *wsparcie* reguły, czyli liczbę przypadków w oryginalnej bazie danych ją potwierdzających.

Rysunek 7.4: Format bazy wiedzy programu RSES

```
RULE_SET wine
ATTRIBUTES 14
 attr0 numeric 6
 attr1 numeric 2
 attr2 numeric 2
 attr3 numeric 2
 attr4 numeric 2
 attr5 numeric 2
 attr6 numeric 2
 attr7 numeric 2
 attr8 numeric 2
 attr9 numeric 2
 attr10 numeric 2
 attr11 numeric 2
 attr12 numeric 2
 class numeric 0
DECISION_VALUES 3
1
2
3
RULES 115
(attr5=300)=>(class=1[6]) 6
(attr0=1208)=>(class=2[5]) 5
(attr10=57)=>(class=3[5]) 5
(attr5=295)&(attr10=125)=>(class=1[3]) 3
(attr4=11800)=>(class=1[3]) 3
(attr12=128500)=>(class=1[3]) 3
(attr4=8800)&(attr0=1237)=>(class=2[3]) 3
(attr4=8600)&(attr7=30)=>(class=2[3]) 3
(attr0=1242)=>(class=2[3]) 3
(attr6=203)=>(class=2[3]) 3
(attr1=161)=>(class=2[3]) 3
(attr2=192)=>(class=2[3]) 3
(attr3=1600)&(attr1=168)=>(class=1[2]) 2
(attr1=173)&(attr10=112)=>(class=1[2]) 2
(attr5=280)&(attr10=104)=>(class=1[2]) 2
(...)
```

Po wczytaniu bazy danych system dokonuje jej analizy w celu zaimportowania danych do wewnętrznego formatu danych i poddania ich grupowaniu. W głównym okienku programu wyświetlane są informacje o wartościach atrybutów poszczególnych reguł (każda reguła w osobnej linijce - patrz rys. 7.5). Oprócz tego, dynamicznie wypełniana jest część formatki o nazwie *Pytanie* zgodnie z nazwami atrybutów wchodzących w skład bazy danych. Wszystkie nieokreślone wartości atrybutów otrzymują wartość *UND* (ang. *undefined*).



Rysunek 7.5: Okienko programu po wczytaniu bazy danych

Przed rozpoczęciem procesu grupowania użytkownik ma możliwość zmiany parametrów grupowania. Korzystając z oznaczeń na rysunku 7.5 mamy:

1. Parametr T określający liczbę grup dla grupowania algorytmem *mAHC*. Wartość ta jest wyznaczana poprzez algorytm omówiony w rozdziale 5.3.3. Po jej wyznaczaniu, zarówno pasek wyboru jak i pole wielokrotnego wyboru (oba oznaczone numerem 1 na rysunku) przyjmują wartość T. Wartość ta może być ręcznie modyfikowana przez użytkownika. Jej liczbowa wartość przedstawiona jest w miejscu oznaczonym liczbą 9 na rys. 7.5.

2. Parametr określający głębokość wyszukiwania hierarchicznego dla algorytmu *AHC*. System umożliwia przeszukanie struktury drzewiastej w celu odnalezienia jednej reguły bądź też skupienia reguł. Informacja o głębokości wyszukiwania, a co za tym idzie – przybliżonej liczności grupy odnajdywanej przez algorytm, jest również podawana przez użytkownika. Dzięki temu parametrowi możliwe jest sterowanie dokładnością wnioskowania.

3. Sposób tworzenia reprezentantów. Parametr ten wybiera typ tworzonego reprezentanta. Szczegółowe informacje nt. obu analizowanych typów znajdują się w rozdziale 5.2.2.

4. Sposób wyliczania macierzy nierozróżnialności (podobieństwa). Parametr ten wskazuje na sposób wyliczania podobieństwa pomiędzy regułami. Dostępne opcje (*simpleSimilarity* oraz *weightedSimilarity*) zostały omówione we wcześniejszym rozdziale dotyczącym miar odległości (5.3.3).

5. Część służąca do ustawiania faktów znanych na początku wnioskowania. Po wczytaniu bazy danych w tej części wyświetlane są wszystkie atrybuty wchodzące w skład reguł w bazie wiedzy. Przed rozpoczęciem procesu wnioskowania, użytkownik może ustawić wartości znanych atrybutów, które staną się faktami.

6. Metoda wyboru ścieżki najbardziej obiecującej dla algorytmu *AHC*. Parametr ten służy do ustawienia metody wyboru węzła najbardziej obiecującego. Dostępne ustawienia odpowiadają oznaczeniom funkcji f_{sim} omówionym w rozdziale 5.3.4.

7. Parametr ustawiający jeden z trzech zaimplementowanych kryteriów łączenia skupień: miarę pojedynczego wiązania, całkowitego wiązania oraz średniego wiązania. Metody te zostały omówione w rozdziale 5.2.4.

8. Wartość współczynnika progowego t_{prec} do wyznaczania optymalnej liczby skupień omówionego w rozdziale 5.3.3. Wartość ta początkowo ustawiona jest na 0,85 ze względu na wyniki eksperymentów przedstawionych w rozdziale 8.4.

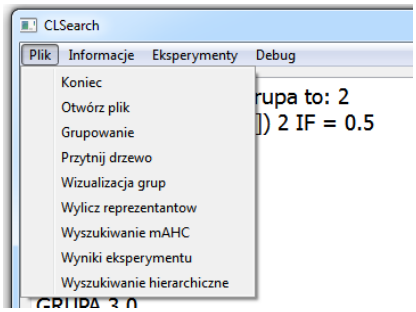
Po procesie wczytania bazy wiedzy, użytkownik powinien wykonać grupowanie wybierając w tym celu opcję *Grupowanie* w celu stworzenia skupień zgodnie z podanymi parametrami. Aby lepiej odróżniać reguły od siebie,

wartości atrybutów decyzyjnych są również brane pod uwagę przy wyliczaniu podobieństwa pomiędzy regułami.

Proces ten kończy tworzenie hierarchicznej bazy wiedzy, która od tej pory rezyduje w pamięci operacyjnej komputera gotowa do dalszej analizy.

7.3 Wyszukiwanie reguł w hierarchicznej bazie wiedzy

Wyszukiwanie reguł przy użyciu autorskiego systemu może zostać przeprowadzone zarówno w pełnej strukturze hierarchicznej wygenerowanej za pomocą algorytmu *AHC*(*Wyszukiwanie hierarchiczne*), jak również przy użyciu reprezentantów grup zgodnie z poziomem przycięcia drzewa wyznaczonych algorytmem *mAHC* (*Wyszukiwanie mAHC*). Wybór jednej z tych dwóch metod dokonywany jest za pomocą etykiet akcji znajdujących się w menu *Plik*.



Rysunek 7.6: Menu Plik z dostępnymi opcjami

7.3.1 Wyszukiwanie reguł przy użyciu algorytmu mAHC

Po dokonaniu procesu grupowania należy przeprowadzić proces przycięcia drzewa na wybranym poziomie. System automatycznie ustawia optymalny poziom przycięcia drzewa (parametr t_{prec}), który może zostać zmieniony przez użytkownika końcowego. Proces przycięcia drzewa opisany jest w podrozdziale 5.3.3.

W podobny sposób możliwe jest przycięcie drzewa w przedstawianym systemie. Po wybraniu odpowiedniej opcji z menu *Plik* w głównym oknie programu wyświetlany jest rezultat tej funkcjonalności:

```
0 0 1
```

```
1 1 0
2 2 16
3 3 14
4 4 108
5 5 104
6 6 106
7 7 656
8 8 99
9 9 358
10 10 654
(...)
```

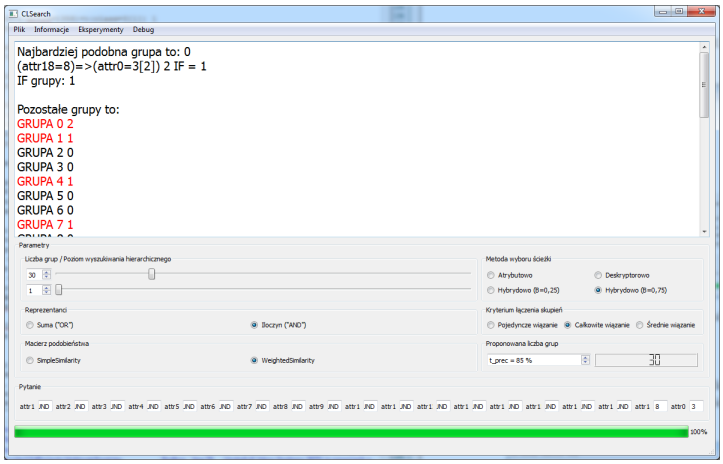
Kolejne liczby oznaczają: kolejny numer reguły, alias numeru reguły (w przypadku gdyby numeracja reguł zawierała wartości puste) oraz numer grupy, do której zostaje przydzielona dana reguła.

Po procesie przycięcia drzewa możliwym jest podgląd struktury poszczególnych grup:

```
(...)
GRUPA 45
(attr3=2450)&(attr0=1358)=>(class=3[1]) 1
(attr0=1358)&(attr1=166)=>(class=1[1]) 1
GRUPA 46
(attr4=10600)&(attr0=1270)=>(class=3[1]) 1
(attr0=1270)&(attr1=387)=>(class=2[1]) 1
GRUPA 47
(attr4=9600)&(attr0=1253)=>(class=3[1]) 1
(attr4=9600)&(attr0=1439)=>(class=1[1]) 1
GRUPA 48
(attr3=1950)&(attr0=1182)=>(class=2[1]) 1
(attr7=37)&(attr0=1237)=>(class=2[1]) 1
(attr4=8800)&(attr3=1950)=>(class=3[2]) 2
(attr4=8800)&(attr12=56200)=>(class=2[2]) 2
(attr4=8800)&(attr0=1237)=>(class=2[3]) 3
(...)
```

W wyniku podglądu struktury grup widoczne są oryginalne reguły wczytane z płaskiej bazy wiedzy. W celu przeprowadzenia wyszukiwania reguły (skupienia) koniecznym jest przeprowadzenie procesu wyliczania reprezentantów poszczególnych skupień, co umożliwia kolejna opcja z menu *Plik*.

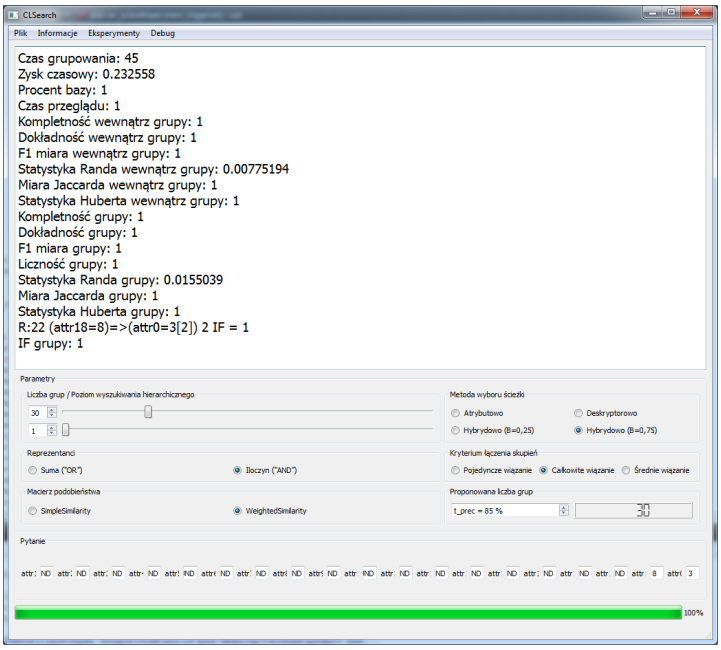
Po zadaniu przykładowego pytania w głównym oknie programu wyświetlane są wartości podobieństwa zbioru faktów do poszczególnych grup w systemie. Grupa o największej wartości podobieństwa zostaje odnaleziona i dla wszystkich reguł w tej grupie zostają wyliczone wartości współczynnika IF[†]. Wynik działania tego modułu programu widoczny jest na rys. 7.7. Możliwym jest również wyświetlenie zaawansowanych statystyk za pomocą opcji *Wyniki eksperymentu* z menu *Plik*. Wynik takiego działania obrazuje rys. 7.8.



Rysunek 7.7: Wynik działania wyszukiwania z użyciem skupień wyznaczonych algorytmem mAHC

Całościowy eksperyment dostarcza dodatkowo informacji o czasie grupowania (liczonym w milisekundach), zysku czasowym (liczonym jako iloraz liczby grup do liczby reguł w systemie), procencie przeglądanej bazy w celu odnalezienia relewantnej reguły (skupienia) oraz omówionych wcześniej parametrach kompletności i dokładności. Efektywność liczona jest zarówno dla grupy jako całości (w tym przypadku liczona w stosunku do reprezentanta grupy), jak również jako średnia kompletności i dokładności reguł wchodzących w skład wybranego skupienia. Wyliczane są również wartości współczynnika IF dla poszczególnych reguł oraz średnia wartość współczynnika IF dla grupy. Dodatkowo, wyliczane są wartości innych parametrów efektywności, takich jak F1-miara, statystyka Randa, miara Jaccarda, Γ statystyka Huberta omówione w rozdziale 6.1.2 niniejszej rozprawy.

[†]Zarówno dla reguł wchodzących w jej skład, jak również średnia wartość współczynnika IF dla całej grupy.



Rysunek 7.8: Wynik eksperymentu wyszukiwania z użyciem skupień wyznaczonych algorytmem mAHC

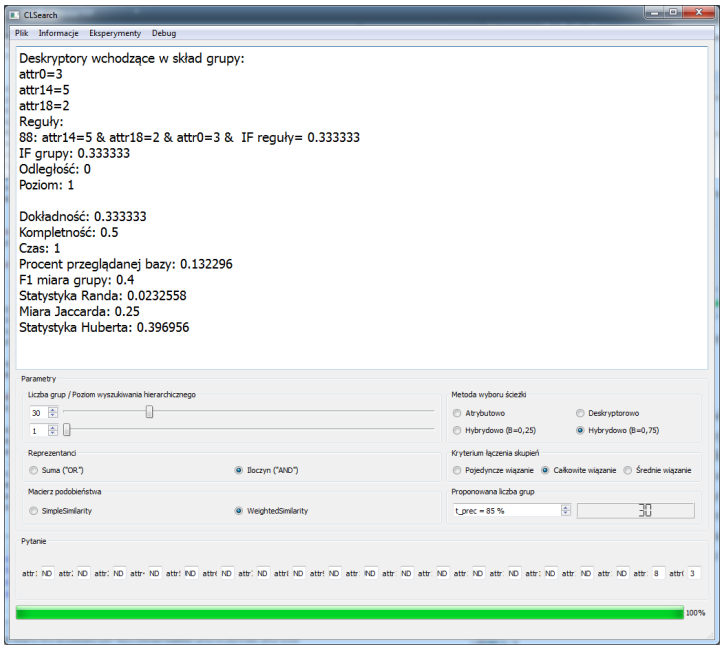
7.3.2 Wyszukiwanie przy użyciu algorytmu AHC

W systemie zaimplementowano również wyszukiwanie reguł (skupień) w pełnej strukturze hierarchicznej (dendrogramie). W tym celu, po wykonaniu grupowania nie trzeba już przycinać drzewa lub też wyliczać reprezentantów skupień.

Po wpisaniu znanych faktów uruchamiane jest *Wyszukiwanie hierarchiczne*, czego efektem jest informacja o strukturze przedstawionej na rysunku 7.9.

Prezentowane są wartości kompletności i dokładności wraz z informacjami o części przeglądanej bazy wiedzy oraz czasie przeglądu (w milisekundach). W wyniku działania systemu przedstawiana jest również struktura wybranej grupy w postaci listy reguł wchodzących w jej skład oraz zbioru deskryptorów wszystkich reguł ją tworzących. Wyliczane są również wartości współczynników IF dla poszczególnych reguł, jak i dla całej grupy oraz odległość pomiędzy skupieniami łączonymi w odnalezionym węźle i poziom znalezionego węzła[‡]. Wyznaczane są także parametry efektywności wyszu-

[‡]Czyli długości ścieżki od korzenia dendrogramu do znalezionego skupienia.



Rysunek 7.9: Wynik eksperymentu wyszukiwania z użyciem struktury hierarchicznej

kiwania takie jak F1-miara, statystyka Randa, miara Jaccarda, Γ statystyka Huberta omówione w rozdziale 6.1.2 niniejszej rozprawy..

Rozdział 8

Eksperymenty obliczeniowe

Celem eksperymentów prowadzonych w ramach rozprawy było zaprojektowanie i implementacja algorytmów wnioskowania dla baz uwzględniających wiedzę niepełną. W tym celu autor zaimplementował dwie wersje algorytmu grupowania reguł (algorytm AHC oraz AHC z modyfikacjami zwany także $mAHC$), opracował model złożonej ze skupień reguł hierarchicznej bazy wiedzy, zaimplementował algorytm propagacji niepełności wiedzy z użyciem autorskiego rozwiązania w postaci współczynników IF, co zostało wykorzystane w algorytmie wnioskowania mogącym być użytym przez dowolny system wspomagania decyzji.

Eksperymenty będą miały na celu wykazanie skuteczności algorytmów grupowania reguł, następnie wykorzystania współczynników IF jako użytecznej metody reprezentacji wiedzy niepełnej, a na końcu ocenę efektywności wnioskowania w przód w bazach wiedzy wykorzystujących proponowane podejście. Źródłem do tworzenia regułowych baz wiedzy było repozytorium Machine Learning [8] i zawarte tam zbiory danych o różnej wielkości i strukturze przechowywanych danych. Proponowane rozwiązanie wspomaga proces wnioskowania i może zostać wykorzystane w celu zbudowania pełnego SWD. Zwiększenie efektywności wnioskowania następuje poprzez znaczne przyspieszenie odnajdywania reguł możliwych do uaktywnienia oraz przez kontrolowane uaktywnianie reguł, których nie wszystkie przesłanki są spełnione.

Eksperymenty służące do ustalenia optymalnych parametrów algorytmu grupującego prowadzone były na dwóch bazach o skrajnie różnej złożoności (Wine oraz Abalone). Po ustaleniu optymalnych parametrów grupowania, autor przystąpił do ich weryfikacji i walidacji na większym zestawie baz

(Wine, Lymphography, Spect, Balance, Tic-Tac-Toe, Car, KRvsKP). Charakterystyki baz danych ujęte są w tabeli 8.1.

Bazy danych z repozytorium zostały wczytane do pakietu RSES w którym za pomocą algorytmu LEM2 [50] wyindukowano reguły minimalne stanowiące wejściową bazę reguł. Na tych bazach reguł wykonywane zostały wszystkie eksperymenty przedstawione w niniejszym rozdziale.

Wyniki przedstawianych badań zostały opublikowane w recenzowanych czasopismach i przedstawione na konferencjach naukowych krajowych i zagranicznych.

Plan przeprowadzonych eksperymentów jest następujący:

Eksperyment nr 1: Analiza wpływu metody tworzenia reprezentanta na jakość skupień. Wyniki tych badań opublikowane zostały w [160].

Eksperyment nr 2: Analiza wpływu miary odległości pomiędzy skupieniami na ich wzajemne rozróżnianie i jakość wnioskowania. Wyniki tych badań opublikowane zostały w [107].

Eksperyment nr 3: Analiza kryterium łączenia skupień w kontekście jakości generowanych skupień. Wyniki tych badań opublikowane zostały w [70].

Eksperyment nr 4: Analiza proponowanej metody szacowania kryterium stopu dla algorytmu $mAHC$. Wyniki tych badań opublikowane zostały w [108].

Eksperyment nr 5: Analiza porównawcza metod modyfikacji algorytmu ścieżki najbardziej obiecującej i ich wpływ na jakość wnioskowania. Wyniki tych badań opublikowane zostały w [71].

Eksperyment nr 6: Analiza porównawcza wnioskowania przy użyciu skupień generowanych przez algorytmy AHC oraz $mAHC$. Wyniki tych badań opublikowane zostały w [108] oraz [71].

Eksperyment nr 7: Analiza liczby możliwych do przeprowadzenia wnioskowań w kontekście różnych stopni niepełności wiedzy (IF). Wyniki tych badań opublikowane zostały w [109] oraz w [72].

Eksperyment nr 8: Analiza możliwości wykorzystania współczynnika IF jako miary niepełności wiedzy. Wyniki tych badań opublikowane zostały w [162].

Tabela 8.1: Parametry baz danych użytych do przeprowadzenia eksperymentów do wyznaczenia optymalnych parametrów algorytmu grupującego

	Wine	Abalone	Lympho-graphy	Spect	Balance	Tic-Tac-Toe	Car	KRvsKP
Liczba atrybutów	14	8	18	23	5	10	7	37
Liczba obiektów	178	4177	148	267	625	958	1728	3196
Liczba wygenerowanych reguł	115	3079	129	193	502	760	781	1152
Rodzaj atrybutów	Ilościowe	Jakościowe, ilościowe	Jakościowe	Jakościowe	Jakościowe	Jakościowe	Jakościowe	Jakościowe
Krótka charakterystyka	Określenie rodzaju wina na podstawie analizy chemicznej.	Określenie wieku małży na podstawie jej parametrów fizycznych.	Klasyfikacja onkologiczna chorych po limfografii.	Klasyfikacja choroby serca na podstawie badania SPECT.	Problem równowagi wagi laboratoryjnej.	Model gry w kółko i krzyżyk.	Ocena zakupu samochodu osobowego.	Ewaluacja partii szachowej.

8.1 Wybór metody tworzenia reprezentanta skupień

Pierwszym problemem w adaptacji algorytmu grupującego jest wybór techniki tworzenia reprezentanta stojącego na czele grupy w przypadku korzystania z algorytmu *mAHC*. Po dokonaniu szczegółowej analizy do dalszych eksperymentów autor wykorzystuje metody tworzenia reprezentanta jako zbioru cech charakteryzujących wszystkie obiekty danej grupy (reprezentant tworzony za pomocą spójnika logicznego "AND") oraz jako zbiór wartości atrybutów opisujących wszystkie obiekty wchodzące w skład danej grupy (typu "OR"). Reprezentant tworzony za pomocą cech unikalnych dla reguł wchodzących w skład danego skupienia okazał się niewystarczający ze względu na fakt gubienia informacji kluczowych dla procesu wnioskowania. Dominujące cechy skupień były w tej metodzie tracone na rzecz rzadko występujących pojedynczych deskryptorów. Z kolei reprezentant jako zbiór cech dominujących tworzył centroidy zbyt podobne do siebie, co w późniejszym procesie wnioskowania powodowało błędny wybór ścieżki najbardziej obiecującej.

Przedmiotem eksperymentu jest pomiar parametrów kompletności (K) oraz dokładności (D) wyszukiwania reguł relewantnych w procesie wnioskowania względem obserwacji początkowych (faktów). Podawana jest także liczność (L) reguł wchodzących w skład odnalezionego skupienia. Wszystkie porównania w obrębie eksperymentu przeprowadzane są w stosunku do reprezentanta grupy.

Poszukiwanie optymalnych parametrów algorytmu grupującego wykonane zostało za pomocą algorytmu *mAHC*. Punkt odcięcia dendrogramu ustawiony został dla bazy Wine na 11 grup, a dla bazy Abalone na 56 skupień. Wartość ta stanowi przybliżenie w postaci pierwiastka kwadratowego z liczby reguł wchodzących w skład bazy wiedzy. W przypadku tego eksperymentu, parametry określające głębokość wyszukiwania hierarchicznego oraz wybór metody wyznaczania ścieżki najbardziej obiecującej nie są wymagane. Wszystkie testy zostały wykonane dla wszystkich kombinacji metod wyznaczania macierzy nierozróżnialności oraz kryteriów łączenia skupień. W tabeli 8.2 przedstawione są maksymalne uzyskane wartości.

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

Oczywiście wiąże się z tym fakt, że parametry kompletności i dokładności nie będą wtedy przyjmować wartości maksymalnych.

W rozprawie zbadano efektywność systemu dla wybranych dwóch metod tworzenia reprezentanta poprzez wykonanie następujących testów:

Test nr 1 zakładał istnienie przynajmniej jednej takiej reguły w bazie wiedzy, której wszystkie przesłanki są prawdziwe (są faktami w bazie wiedzy). Podając zatem wśród obserwacji te wszystkie deskryptory, które występują w części warunkowej i decyzyjnej przynajmniej jednej reguły, starano się odnaleźć grupę zawierającą tę regułę.

Test nr 2 Do początkowo pustego zbioru faktów dopisano losowo wybrane deskryptory spośród wszystkich obecnych w systemie. Aby sprawdzić skuteczność radzenia sobie systemu z dużą dozą losowości, wykonano test, w którym system miał odnaleźć najbardziej relewantną grupę w stosunku do całkowicie losowego zbioru deskryptorów.

Test nr 3 Zbiór faktów składa się z 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.

Test nr 4 Dokładnie jedna para atrybut-wartość stanowi cały zbiór faktów. Sprawdzono również, czy system poradzi sobie z pytaniem ogólnym, zawierającym jeden deskryptor użyty w systemie. Spodziewaną odpowiedzią jest dość liczny zbiór reguł.

Wyniki eksperymentów przedstawione są w tabeli 8.2.

Jak widać, reprezentant typu "AND" uzyskiwał znacznie lepsze wyniki kompletności przy stosunkowo dużych wartościach parametru dokładności. Widać również, iż otrzymane w wyniku wnioskowania grupy w tym przypadku były mniej liczne ze względu na znacznie lepszą odróżnialność ich od siebie. 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).

Reprezentant „AND" ma tendencję do tworzenia małorozróżnialnych grup o podobnym opisie, natomiast reprezentant „OR" tworzy bardzo długie opisy, trudne w dalszym przetwarzaniu. Nie sprawdziła się koncepcja

	Reprezentant "AND"			**Reprezentant"OR"**		
Nr testu	K	D	L	K	D	L
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

Tabela 8.2: Eksperyment nr 1: Wybór metody tworzenia reprezentanta grupy.

umieszczania w reprezentancie unikalnych w skali całego systemu deskryptorów reguł wchodzących w skład grupy.

Stwierdzono również w wyniku eksperymentów, iż stosunkowo częstą jest sytuacja gdy kilka grup jest tak samo podobnych do zadanego pytania (sytuacja ma zwłaszcza miejsce 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.

8.2 Wybór metody miary odległości pomiędzy grupami

Drugą kwestią badaną w rozprawie był wybór miary odległości pomiędzy dwoma regułami. Metoda prostego podobieństwa i ważonego podobieństwa zostały omówiona w rozdziale 5.3.3.

Podobnie jak w poprzednim eksperymencie, tutaj również skorzystano z algorytmu *mAHC* z wstępnie ustawioną przybliżoną liczbą skupień (11 dla bazy Wine, 56 dla bazy Abalone). Również w przypadku tego eksperymentu, parametry określające głębokość wyszukiwania hierarchicznego oraz wybór metody wyznaczania ścieżki najbardziej obiecującej nie są wymagane. Wszystkie testy zostały wykonane dla wszystkich kombinacji metod

wyznaczania reprezentanta oraz kryteriów łączenia skupień. W tabeli 8.3 przedstawione są maksymalne uzyskane wartości.

Wyniki eksperymentu przedstawia tabela 8.3.

	Proste podobieństwo			Ważone podobieństwo		
Nr testu	K	D	L	K	D	L
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	0

Tabela 8.3: Eksperyment nr 2: Wybór miary odległości pomiędzy regułami.

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

Jakkolwiek wyniki dokładności były lepsze dla prostego podobieństwa, tak (zwłaszcza dla bazy Abalone) wyniki kompletności dla ważonego podobieństwa były znacznie lepsze dla podobieństwa prostego przy wystarczająco dobrych wynikach dokładności. Podobnie jak w przypadku eksperymentów dotyczących metody wyznaczania reprezentanta, tak i tutaj mamy do czynienia z lepszymi rezultatami dla podobieństwa typu ważonego. Powody tego stanu rzeczy są takie same jak przytoczone przy omawianiu wyników poprzedniego eksperymentu.

Metoda prostego podobieństwa daje zdecydowanie lepsze rezultaty 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.

8.3 Wpływ kryterium łączenia skupień

Problemem wartym zbadania jest także wpływ kryterium łączenia skupień na ich jakość. W tym celu postanowiono przeprowadzić testy analogiczne do przedstawionych powyżej. Do grupowania użyto trzech miar łączenia skupień: całkowitego wiązania (CL), średniego wiązania (AL) i pojedynczego wiązania (SL).

Parametry eksperymentu były analogiczne do poprzednich: algorytm *mAHC* z 11 skupieniami dla bazy Wine, 56 dla bazy Abalone, do tabeli wynikowej wpisywana wartość maksymalna dla wszystkich kombinacji metod wyznaczania reprezentanta oraz miary odległości pomiędzy grupami.

Wyniki eksperymentów przedstawia tabela 8.4.

	SL			AL			CL		
Nr testu	K	D	L	K	D	L	K	D	L
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

Tabela 8.4: Eksperyment nr 3: Wybór kryterium łączenia skupień.

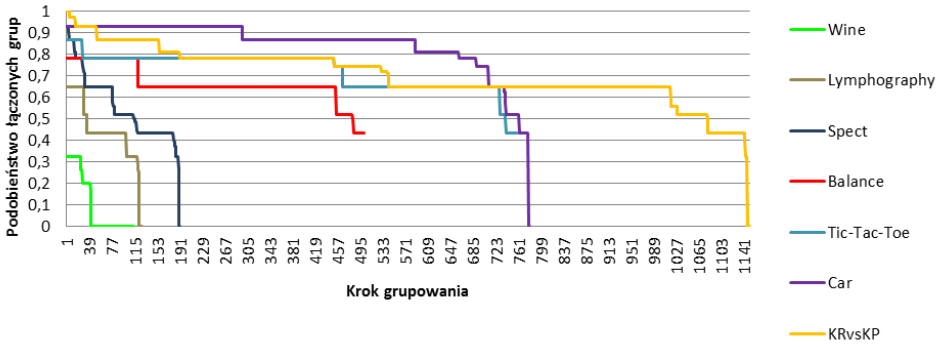
Wyniki przeprowadzonych eksperymentów dotyczących kryterium wiązania reguł w grupy dla bazy Wine nie pozwoliły na jednoznaczne rozstrzygnięcie, która z metod da lepsze rezultaty. Dopiero eksperymenty na znacznie większej bazie Abalone rozstrzygnęły jednoznacznie, iż to metoda całkowitego wiązania będzie bardziej efektywna.

Przed rozpoczęciem eksperymentów zakładano, 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. Dzięki temu uzyskujemy dużą liczbę małolicznych grup.

8.4 Proponowane kryterium stopu dla ustalenia liczby skupień w danych

W rozdziale 5.3 przedstawiono proponowane kryterium stopu dla algorytmu $mAHC$. Eksperyment w tym zakresie polegał na znalezieniu optymalnego punktu przerwania grupowania uruchamiając algorytm dla kilku różnych baz (czyli wyznaczeniu wartości współczynnika t_{prec}. W każdym kroku grupowania zapisywano wartość wzajemnego podobieństwa łączonych grup. Pozostałe parametry zostały ustalone poprzez analizę wcześniejszych eksperymentów. Reprezentant wyznaczany został za pomocą metody "AND", miarą odległości była metoda ważona, a skupienia łączone zostały za pomocą metody całkowitego wiązania.

Wykres tych wartości przedstawiony jest na rys. 8.1

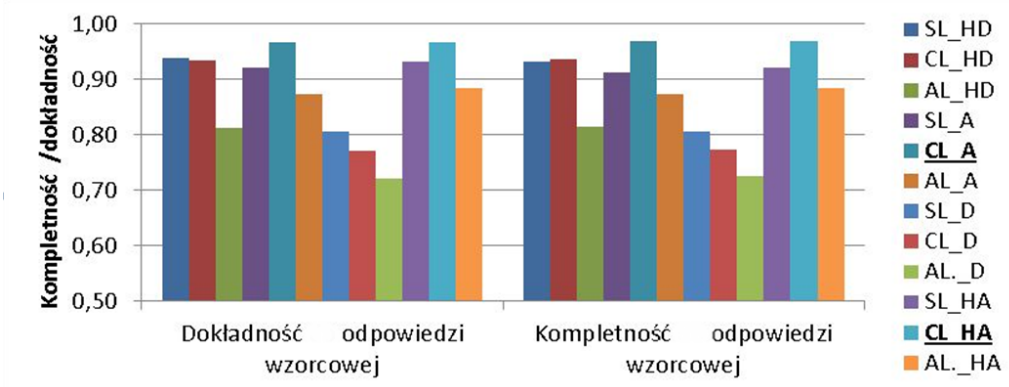


Rysunek 8.1: Eksperyment nr 4: Podobieństwo grup łączonych w poszczególnych krokach algorytmu grupowania.

Widać wyraźnie, że w pewnym momencie jakość grupowania drastycznie spada. Uważa się, że jest to punkt, w którym łączone ze sobą skupienia reguł są już dość mało do siebie podobne. Jest to sygnał do zaprzestania grupowania. W wyniku eksperymentów, najlepsze rezultaty osiąga się w przypadku gdy współczynnik t_{prec} ma wartość $0,85$, co warunkuje z kolei $T = 0,85 \cdot simMax$, gdzie $simMax$ to maksymalna wartość podobieństwa dwóch łączonych reguł w pierwszym kroku grupowania.

8.5 Dobór parametrów do metody ścieżki najbardziej obiecującej

W celu sprawdzenia, która z przedstawionych metod wyboru ścieżki najbardziej obiecującej da lepsze rezultaty, przeprowadzono eksperymenty obliczeniowe zgodnie z przedstawionym scenariuszem. Na początku, przyjmowano, że deskryptory opisujące aktualnie analizowaną regułę stanowią jednocześnie zbiór faktów. Do pełnego systemu wyznaczonego za pomocą różnej kombinacji metod ścieżki najbardziej obiecującej oraz metody łączenia skupień, zadawano pytanie składające się z iloczynu deskryptorów tworzących zbiór faktów. Odpowiedź systemu traktowano jako odpowiedź wzorcową. Następnie, z bazy wiedzy usuwano tę konkretną analizowaną regułę i powtarzano proces wnioskowania dla tego samego zbioru faktów. Sprawdzano następnie wartości kompletności i dokładności wyszukiwania odnalezionej reguły dla takiego przypadku. Na wszystkich wykresach przyjęto następujące oznaczenia: SL – metoda pojedynczego wiązania, CL – metoda całkowitego wiązania, AL – metoda średniego wiązania, HD – miara hybrydowa ścieżki najbardziej obiecującej ze zwiększoną wartością współczynnika dla wspólnych deskryptorów ($B_1 = 0,75$ $B_2 = 0,25$), HA - miara hybrydowa ścieżki najbardziej obiecującej ze zwiększoną wartością współczynnika dla wspólnych atrybutów ($B_1 = 0,25$ $B_2 = 0,75$), A – miara pokrycia atrybutowego, D – miara pokrycia deskryptorowego. Wyniki eksperymentu przeprowadzonego na bazie Balance zawierającej 502 reguły minimalne przedstawione są na rysunku 8.2.

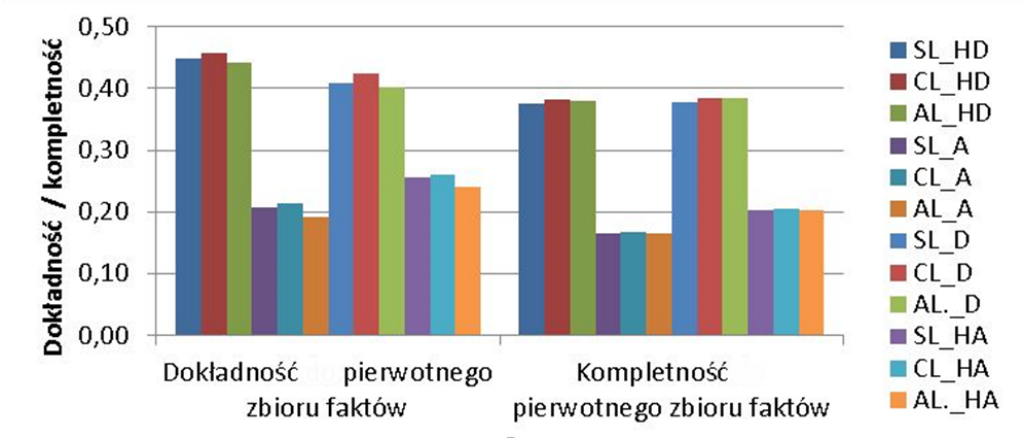


Rysunek 8.2: Eksperyment nr 5: Eksperymenty dla metody ścieżki najbardziej obiecującej.

Najlepsze rezultaty uzyskano gdy algorytm korzystał z metody CL

do wiązania skupień niezależnie od metody wyboru ścieżki. Potwierdza to poprzednie eksperymenty służące wypracowaniu optymalnego kryterium łączenia skupień. W przypadku oceny dokładności i kompletności w stosunku do najlepszej grupy, wyniki wszystkich przedstawionych podejść prezentują zbliżone rezultaty (najlepszą jest miara hybrydowa z większą wagą dla wartości wspólnych atrybutów oraz metody pokrycia atrybutowego). Wydaje się, że podejście rozróżniające grupy daje lepsze rezultaty w stosunku do tradycyjnego podejścia pokrycia deskryptorowego.

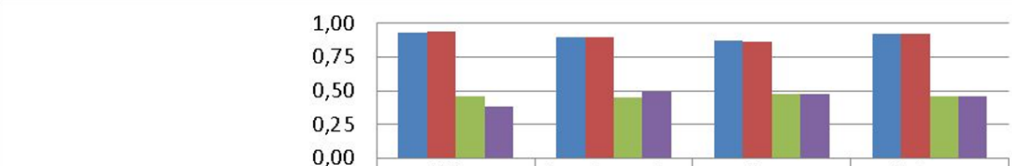
W drugiej części eksperymentu, obliczono kompletność i dokładność odpowiedzi dla ograniczonego systemu w stosunku do pytania składającego się z iloczynu deskryptorów tworzących zbiór faktów. W ten sposób badana jest zdolność stworzonego systemu do kompensowania niepełnej wiedzy. Należy jednakże zauważyć, że wartości kompletności i dokładności bliskie maksymalnym nie mogą wystąpić, ze względu na fakt usunięcia reguły, która jest optymalną odpowiedzią na tak zadane pytanie. Wyniki tego eksperymentu prezentuje rysunek 8.3.



Rysunek 8.3: Eksperyment nr 6: Eksperymenty dla metody ścieżki najbardziej obiecującej.

Rysunek 8.3 jednoznacznie pokazuje, że autorska metoda hybrydowa ze zwiększoną wartością współczynnika dla wspólnych deskryptorów ($B_1 = 0,75$ $B_2 = 0,25$) sprawdza się lepiej, niezależnie nawet od sposobu łączenia skupień. Wartości parametrów efektywności uzyskiwane dla tej metody są blisko dwukrotnie większe od pozostałych rozwiązań. Do dalszych eksperymentów przyjęto metodę całkowitego wiązania (CL) oraz ścieżki najbardziej obiecującej ze wzmocnieniem wag dla wspólnych deskryptorów (HD).

W celu walidacji proponowanego rozwiązania wraz z ustalonymi parametrami, przeprowadzono eksperymenty dla większej liczby baz wiedzy. Wyniki prezentuje rys. 8.4. Bliskie maksymalnym wartości kompletności i dokładności w stosunku do grupy optymalnej potwierdzają słuszność zaproponowanego podejścia w problemie wnioskowania w systemach z wiedzą niepełną.



	Wine	Lymphography	Spect	Balance
Dokładność odpowiedzi wzorcowej	0,93	0,89	0,87	0,92
Kompletność odpowiedzi wzorcowej	0,94	0,89	0,87	0,92
Dokładność pierwotnego zbioru faktów	0,46	0,45	0,47	0,46
Kompletność pierwotnego zbioru faktów	0,38	0,49	0,47	0,46

Rysunek 8.4: Eksperyment nr 7: Walidacja wyznaczonych parametrów metody ścieżki najbardziej obiecującej.

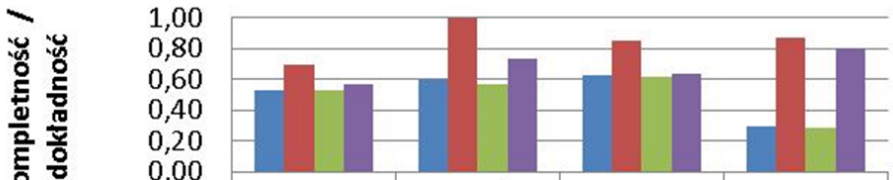
8.6 Wnioskowanie z użyciem reprezentanta a wnioskowanie z użyciem struktury hierarchicznej

Algorytmy AHC oraz $mAHC$ pozwalają na przeprowadzenie wnioskowania w dwojaki sposób. W pierwszym przypadku generowana jest struktura hierarchiczna, gdzie za pomocą metody ścieżki najbardziej obiecującej odnajdowana jest grupa reguł relewantnych do aktualnego zbioru faktów. Reguły te zostają uaktywnione, a wiedza przez nie dostarczona – dopisana do bazy faktów z odpowiednią wartością współczynników IF. Przykład tego wnioskowania przedstawiony jest w rozdziale 5.3.5 na stronie 133. Algorytm $mAHC$ przerywa grupowanie w pewnym ściśle określonym momencie dzięki przedstawianemu tu kryterium stopu. Utworzony zostaje reprezentant (centroid) dla każdej z rozłącznych grup. Wnioskowanie sprowadza się do odnale-

zienia reprezentanta najbardziej relewantnego w stosunku do zbioru faktów i uaktywnienia reguł wchodzących w skład odnalezionego skupienia.

W celu porównania obu tych metod, testowano je dla czterech wybranych baz wiedzy. Przygotowano dla każdej z nich osobno po 10 zestawów faktów losowo wybranych spośród przesłanek i konkluzji reguł faktycznie zapisanych w tych bazach. Obliczono wartości średniej kompletności [rozumianej tutaj jako stosunek liczby wspólnych deskryptorów występujących zarówno w zbiorze faktów jak i w reprezentancie ($mAHC$) lub otrzymanej grupie (AHC) do liczby wszystkich deskryptorów opisujących zbiór faktów i reprezentanta ($mAHC$) lub grupę (AHC)]. Obliczano również średnią dokładność w sposób analogiczny. Dzięki wynikom poprzednich eksperymentów, możliwym było skorzystanie z optymalnych parametrów algorytmu grupującego (ważona miara podobieństwa, kryterium łączenia skupień – miara całkowitego wiązania, reprezentant typu "AND", parametr $T_{prec} = 0,85$).

Wyniki eksperymentów przedstawione są na rysunkach 8.5 oraz 8.6.

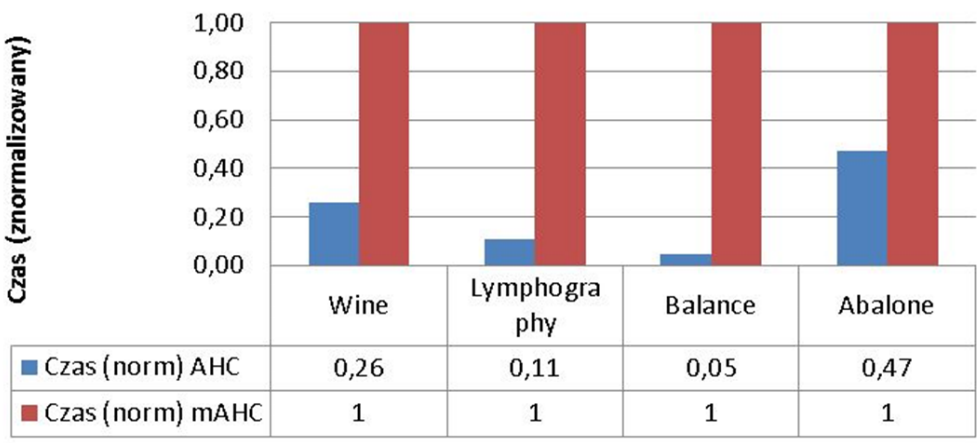


	Wine	Lympho graphy	Balance	Abalone
Kompletność AHC	0,53	0,60	0,63	0,29
Kompletność mAHC	0,70	1,00	0,85	0,87
Dokładność AHC	0,53	0,57	0,62	0,29
Dokładność mAHC	0,57	0,74	0,64	0,80

Rysunek 8.5: Eksperyment nr 8: Jakość wnioskowania hierarchicznego i z użyciem reprezentanta.

Niestety wyszukiwanie z użyciem hierarchii daje minimalnie gorsze rezultaty od podejścia z użyciem reprezentanta. Sugeruje to konieczność dalszych prac nad udoskonaleniem tego podejścia.

Przeanalizowano również czas wyszukiwania reguł do uaktywnienia dla obu algorytmów. Oczywistym jest, że podejście hierarchiczne będzie znacznie szybsze i hipoteza ta znalazła swoje potwierdzenie w wynikach ekspe-



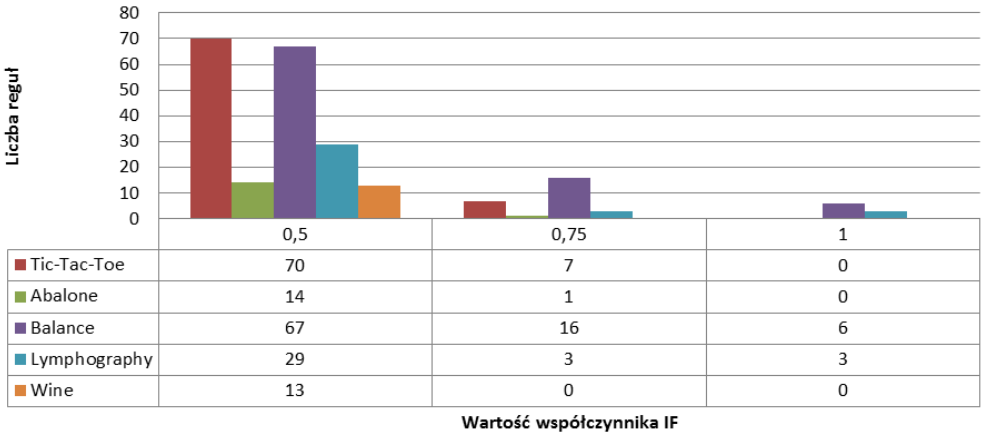
Rysunek 8.6: Eksperyment nr 9: Czas wnioskowania z użyciem algorytmów AHC i $mAHC$.

rymentalnych. Co więcej — im baza reguł większa, tym widoczny większy zysk czasowy dla podejścia hierarchicznego.

8.7 Liczba możliwych przeprowadzonych wnioskowań a wartość współczynnika IF

Aby zaprezentować przydatność opisywanego podejścia, zbadano również liczbę możliwych do przeprowadzenia wnioskowań w zależności od minimalnej progowej wartości współczynnika IF. W zaproponowanym eksperymencie dla każdej z testowanych baz, wylosowano około 10% deskryptorów występujących w całym systemie. Następnie deskryptory te oznaczono jako fakty. Sprawdzono pokrycie reguł w bazie wiedzy takim zbiorem faktów, czyli dokonano analizy liczby reguł możliwych do uaktywnienia przy losowo wybranym zbiorze faktów. Analizę tę dokonano w rozróżnieniu minimalnego współczynnika IF pozwalającego na uaktywnienie danej reguły (innymi słowy: dla $IF = 0,75$ tylko 75% przesłanek musiało być spełnionych, aby uznać regułę za możliwą do uaktywnienia). Eksperyment przeprowadzono bez wykonywania grupowania, stąd wszystkie parametry algorytmu grupującego nie musiały być ustawiane. Wyniki przedstawia rys. 8.7.

Jak widać, jeśli dopuszczone zostanie wnioskowanie tylko na regułach, których wszystkie przesłanki są spełnione (a więc wartość współczynnika IF wynosi 1), to aż w trzech przypadkach nie otrzymamy żadnej nowej wiedzy. Jeśli tylko obniżymy minimalny próg współczynnika IF do wartości 0,75



	0,5	0,75	1
Tic-Tac-Toe	70	7	0
Abalone	14	1	0
Balance	67	16	6
Lymphography	29	3	3
Wine	13	0	0

Rysunek 8.7: Eksperyment nr 10: Liczba możliwych do uaktywnienia reguł a minimalny współczynnik IF.

system będzie w stanie zwrócić informacje, które mogą posłużyć do uszczegółowienia zapytania, a co za tym idzie – do skutecznego wspomagania decyzji podejmowanej przez człowieka-eksperta.

8.8 Współczynnik IF jako miara niepełności wiedzy

Przeprowadzono również testy skuteczności wnioskowania w oparciu o przykładowe bazy wiedzy. W pierwszym z eksperymentów do systemu zawierającego hierarchicznie uporządkowaną bazę wiedzy zadano pytanie składające się z iloczynu deskryptorów tworzących zbiór faktów. W odpowiedzi system zwracał grupę reguł uznawanych za najbardziej relewantne spośród wszystkich reguł obecnych w systemie. Wewnątrz tej grupy reguł odszukiwano taką, której współczynnik IF w stosunku do zadanego pytania był maksymalny. Pozostałe parametry algorytmu *AHC* ustawione zostały analogicznie do eksperymentu przedstawionego w rozdziale 8.6.

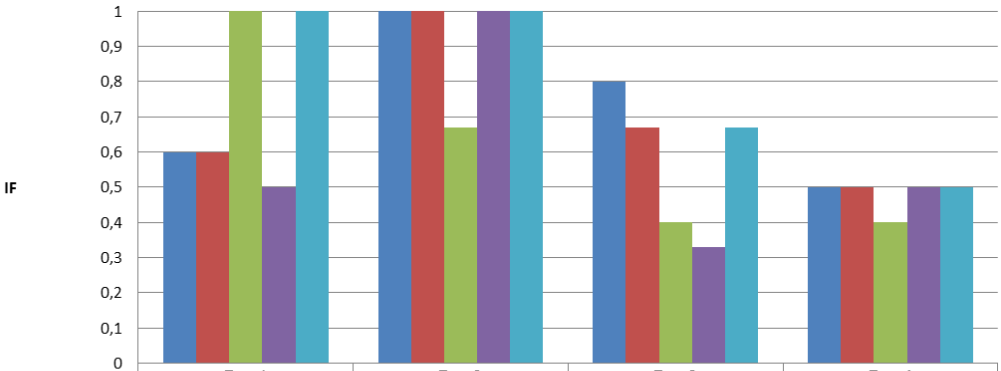
Test został podzielony na cztery przypadki testowe klasyfikowane następująco:

1. T1 oraz T2 – baza faktów składała się ze wszystkich przesłanek losowo wybranej reguły z bazy.

2. T3 — baza faktów to około 80% przesłanek losowo wybranej reguły.

3. T4 — baza faktów to około 50% przesłanek losowo wybranej reguły.

Wyniki eksperymentu zaprezentowane są na rys. 8.8.



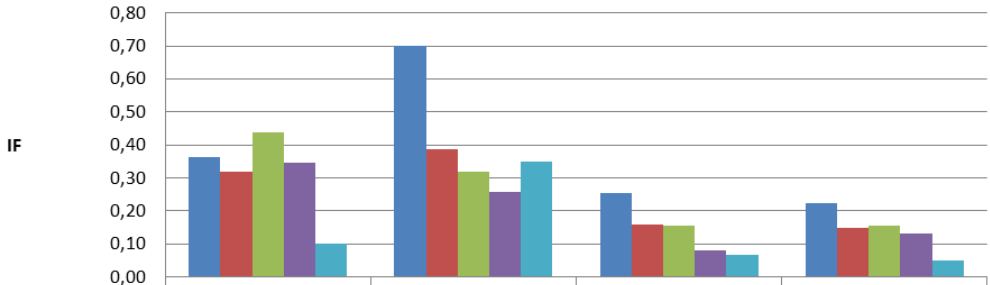
	Test 1	Test 2	Test 3	Test 4
Tic-Tac-Toe	0,6	1	0,8	0,5
Abalone	0,6	1	0,67	0,5
Balance	1	0,67	0,4	0,4
Lymphography	0,5	1	0,33	0,5
Wine	1	1	0,67	0,5

Rysunek 8.8: Eksperyment nr 11: Maksymalna wartość współczynnika IF wewnątrz odnalezionego skupienia.

Wyniki eksperymentu pozwalają sądzić, iż proponowany system dobrze radzi sobie z odnajdowaniem grup reguł relewantnych w stosunku do zadawanego zbioru faktów. Należy tutaj podkreślić, że w teście nr 3 maksymalna możliwa wartość współczynnika IF wynosi około 0,8, a w teście 4 – około 0,5. Jak widać, algorytm daje rezultaty zbliżone do najlepszych możliwych w większości przypadków. Dla większości baz udało się znaleźć reguły najbardziej odpowiednie do uaktywnienia.

Kolejny z eksperymentów miał na celu zbadanie średniej wartości współczynnika IF wewnątrz odnalezionej grupy. Sposób przeprowadzenia eksperymentu był analogiczny do poprzedniego, z tą różnicą, iż badano tutaj średnią wartość współczynnika IF wewnątrz odnalezionej grupy. Dzięki temu możliwym jest zbadanie, czy system generuje skupienia o wysokiej jakości, w których reguły są spójne i mają dużą część wspólnych przesłanek. Wyniki przedstawione są na rys. 8.9.

Otrzymane wyniki pokazują, iż system zwraca stosunkowo liczne skupienia, dzięki czemu średnia wartość współczynnika IF wewnątrz skupienia jest niska. Rezultat ten mówi o konieczności przeglądu otrzymanych skupień przed uaktywnieniem wszystkich reguł w nich się znajdujących. Dzięki temu wiedza otrzymana w wyniku wnioskowania będzie zbliżona do



	Test 1	Test 2	Test 3	Test 4
Tic-Tac-Toe	0,36	0,70	0,25	0,22
Abalone	0,32	0,39	0,16	0,15
Balance	0,44	0,32	0,16	0,16
Lymphography	0,35	0,26	0,08	0,13
Wine	0,10	0,35	0,07	0,05

Rysunek 8.9: Eksperyment nr 12: Średnia wartość współczynnika IF wewnątrz odnalezionego skupienia.

pewnej. Nieefektywnym rozwiązaniem będzie uaktywnianie wszystkich reguł wchodzących w skład odnalezionej grupy.

8.9 Podsumowanie wyników eksperymentów

Zaproponowane rozwiązanie pozwoliło na znaczne zmniejszenie liczby przeglądanych reguł. Zysk czasowy wynikający z możliwości przeglądania jedynie reprezentantów grup reguł zamiast wszystkich reguł jedna po drugiej widoczny jest zwłaszcza dla dużych baz wiedzy. Przykładowo, gdy baza reguł składa się 1000 rekordów, wystarczy przejrzeć $log_2 1999 \approx 11$ elementów (tj. około 1% klasycznej bazy wiedzy). Dla bazy zawierającej 50 reguł, przegląd $log_2 99 \approx 10$ elementów (tj. 20% klasycznej bazy wiedzy), co jest znacznie mniejszym zyskiem. W efekcie, proponowane rozwiązanie pozwala ograniczyć liczbę reguł faktycznie analizowanych w procesie wnioskowania z użyciem reprezentantów do kilku procent (3-10%). Średni zysk czasowy systemu jest związany z liczbą utworzonych grup i jest szacowany na: $\frac{k}{n}$, gdzie k to liczba skupień, a n to liczba reguł w systemie. Zakłada się jednocześnie, że porównanie każdej reguły ze zbiorem faktów zajmuje jedną jednostkę czasu.

W zdecydowanej większości przypadków, co obrazują tabele przedstawiające wyniki eksperymentów, system był w stanie poprawnie odnaleźć regułę na podstawie wszystkich przesłanek ją opisujących 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 pożądanym zjawiskiem.

W wyniku licznych eksperymentów, najbardziej obiecujące wyniki otrzymano dla metody całkowitego wiązania, reprezentantów jako iloczynów deskryptorów wspólnych oraz ważonej miary podobieństwa. Metoda wyznaczania kryterium stopu pozwoliła na udowodnienie hipotezy, że dla dużej liczby małolicznych skupień otrzymujemy skupienia charakteryzuje się dużą wewnętrzną spójnością.

Modyfikacja metody ścieżki najbardziej obiecującej pozwoliła na dalsze skrócenie czasu wnioskowania oraz umożliwiła odnalezienie reguł o minimalnej liczbie niespełnionych przesłanek w krótkim czasie. Aktywacja tych reguł przyczyni się do zwiększenia liczby generowanych faktów, a dzięki temu do wyprowadzenia większej wiedzy z systemu. Wnioskiem z przedstawionych powyżej badań jest ustalenie wartości współczynnika progowego dla wyznaczenia optymalnej liczby grup na około 85% wartości największego podobieństwa dwóch reguł między sobą. Dzięki temu, liczba grup jest stosunkowo duża. Mimo zwiększonego nakładu na wyznaczanie reprezentantów oraz porównywanie zbioru faktów z regułami bądź ich skupieniami, proponowane podejście korzystające z algorytmu $mAHC$ daje wysoką jakość odnajdowanych skupień, a tym samym reguł. Mechanizm wnioskowania korzystający z algorytmu AHC dostosowany został do specyfiki grupowania reguł w bazie wiedzy.

W trakcie eksperymentów autor spotkał się z tendencją do łańcuchowania grup reguł. Krótki opis każdej reguły oraz ich mała rozróżnialność względem siebie mogą przyczynić się do zaburzenia równomierności dendrogramu (zdarzało się, że w jednym z poddrzew na każdym poziomie mamy jedną regułę tylko, a w drugim – pozostałe). Po przeanalizowaniu sytuacji, zauważono niepokojący fakt małej rozróżnialności wartości w macierzy podobieństwa budowanej na początku działania algorytmu. Przykładowo, dla bazy Abalone, liczba komórek macierzy podobieństwa wynosiła 7138531, gdzie dla całej bazy występowały tylko 43 różne wartości podobieństwa reguł (a więc wiele było par tak samo podobnych) i o kolejności tworzenia skupień decydowała kolejność par reguł (lub w późniejszym etapie – skupień reguł) w bazie wiedzy.

Optymistycznym wnioskiem jest poprawa algorytmu wyszukiwania reguł metodą ścieżki najbardziej obiecującej za pomocą autorskiej metody hybrydowej z większą wagą dla wspólnych deskryptorów. Oprócz tego, nastąpiło

dalsze skrócenie czasu wnioskowania poprzez wyszukiwanie reguł relewantnych z użyciem hierarchii.

Dzięki zastosowaniu analizy skupień możliwym jest uzyskanie dodatkowej wiedzy z systemu w przypadku wystąpienia impasu, czyli braku reguł możliwych do uaktywnienia w klasycznym przypadku. Proponowane rozwiązanie pozwala na uaktywnianie reguł pewnych (przy wiedzy pełnej) oraz reguł niepewnych (przy wiedzy niepełnej) dzięki określeniu współczynnika niepełności wiedzy (IF).

Rozdział 9

Podsumowanie

Systemy wspomagania decyzji znalazły swoje zastosowanie w wielu różnych dziedzinach życia. Fakt lawinowego wzrostu rozmiaru danych, które są przetwarzane oraz powszechność użycia SWD sprawia, że efektywność procesów wnioskowania staje się niezmiernie ważnym polem do optymalizacji. Nie zawsze bowiem udaje się pozyskać eksperta-człowieka, który swym doświadczeniem i wiedzą będzie mógł służyć do rozwiązywania pewnej klasy problemów. SWD sprawiają, że wiedza ekspercka może niejako być powielona i wykorzystana w wielu miejscach jednocześnie.

Najlepszy ekspert oraz najlepszy nawet SWD nie będą mogli wykonać prawidłowo zadań przed nimi stawianych, jeśli wiedza którą dysponują będzie niepełna. Jednakże, w obliczu coraz większych baz danych i baz wiedzy braki wydają się być nieuniknione, stąd równolegle poszukuje się metod, które wspomogą zarówno eksperta, jak i system ekspertowy, w radzeniu sobie z niepełnością wiedzy. Zwiększenie efektywności takich systemów ma miejsce zarówno poprzez szybsze odnalezienie reguł, które są najbardziej adekwatne do posiadanej aktualnie wiedzy oraz możliwość uaktywnienia tych reguł, co do których nie jesteśmy całkowicie przekonani, że są pewne. Aby poprawnie poradzić sobie z tymi problemami, należy zmodyfikować sposób zapisu wiedzy w bazie wiedzy oraz zaproponować nowe sposoby wnioskowania wykorzystujące informacje o niepełności wiedzy. Obserwacje te stanowiły podstawę niniejszej rozprawy.

Aby uporządkować bazę wiedzy autor skorzystał z mechanizmów analizy skupień. Jak potwierdziły to badania i eksperymenty, wybór ten okazał się słuszny i pozwolił na wykazanie słuszności tezy rozprawy przedstawionej na wstępie. Tworzenie skupień reguł okazało się metodą nie tylko na

uporządkowanie reguł i zorganizowanie ich w hierarchiczną bazę wiedzy, ale także na zmodyfikowanie procesu wnioskowania w systemach wspomagania decyzji z wiedzą niepełną. Dzięki temu do analizy nie jest wykorzystywana cała baza wiedzy, a jedynie jej drobny fragment najbardziej relewantny do pytania. Kolejnym niezmiernie ważnym aspektem jest modularyzacja systemu ekspertowego. Podobnie jak w klasycznej wersji, baza wiedzy w przedstawianym systemie stanowi odrębną strukturę danych możliwą do wymiany i uaktualniania. Mechanizmy wnioskowania działają niezależnie od dziedziny wiedzy zapisanej w bazie wiedzy zorganizowanej zgodnie z przedstawianym tu sposobem.

Szczegółowe wyniki rozprawy

Rozprawa przedstawia wyniki zastosowania grupowania reguł w regułowych bazach wiedzy w celu przyspieszenia procesu wnioskowania oraz zwiększenia jakości wnioskowania w bazach z wiedzą niepełną.

Najważniejsze szczegółowe wyniki rozprawy są następujące:

1. Przystosowanie algorytmów analizy skupień do grupowania reguł i utworzenie hierarchicznej bazy wiedzy. Dokonano analizy różnych parametrów algorytmów AHC oraz $mAHC$ w celu dobrania ich optymalnej wartości dla specyficznego problemu grupowania reguł. Zaproponowane rozwiązanie buduje skupienia reguł podobnych do siebie zgodnie z zaproponowaną miarą podobieństwa. Głęboka analiza istniejących rozwiązań pozwoliła na wybranie algorytmów aglomeracyjnego grupowania hierarchicznego AHC oraz $mAHC$ do rozwiązania przedstawionego problemu. Dzięki temu znacząco zmniejszyła się liczba analizowanych reguł w procesie wnioskowania.

2. Przegląd i porównanie istniejących algorytmów wnioskowania w SWD połączone z ich analizą pod kątem wykorzystania we wnioskowaniu w systemach z wiedzą niepełną. Przedstawiono również istniejące metody reprezentacji wiedzy niepełnej wraz z ich krytyczną analizą. W dalszym kroku zaproponowano nowy sposób reprezentacji niepełności wiedzy.

3. Przedstawiono różne metody oceny efektywności algorytmów analizy skupień oraz efektywności wnioskowania. Metody te pozwoliły na przeprowadzenie eksperymentów mających na celu ustalenie optymal-

nych parametrów grupowania reguł (miary podobieństwa, kryterium łączenia skupień, itp.).

4. Zaproponowano szereg modyfikacji metody ścieżki najbardziej obiecującej. Analiza wspólnych deskryptorów oraz atrybutów dla dwóch analizowanych węzłów pozwoliła na dalsze skrócenie czasu potrzebnego do odnalezienia relewantnego skupienia w procesie wnioskowania.

5. Przedstawiono propozycję kryterium stopu dla algorytmu *mAHC* w problemie grupowania reguł. Dzięki obserwacji podobieństwa łączonych skupień na każdym kroku grupowania możliwe było przerwanie tego procesu w odpowiednim momencie.

6. Opracowano metodę określania stopnia niepełności wiedzy (IF), bazując na metodzie współczynników pewności CF, niezbędnej do realizacji procesów wnioskowania w przypadku regułowych baz wiedzy oraz niepełnego zbioru faktów.

7. Dla potwierdzenia tezy rozprawy wykonano szereg eksperymentów, których wyniki oraz analiza stanowią osobny rozdział tej pracy. Eksperymenty prowadzone na danych z ogólnodostępnego repozytorium pokazały, iż przedstawiane rozwiązanie może być uznane za poprawiające efektywność wnioskowania w systemach z wiedzą niepełną. Częściowe wyniki rozprawy zostały zaprezentowane na krajowych i międzynarodowych konferencjach naukowych oraz opublikowane w recenzowanych czasopismach. Wszystkie przedstawiane metody zostały zaimplementowane i przetestowane w autorskim systemie *CLSearch*, którego dokumentacja stanowi osobny rozdział niniejszej rozprawy.

Bibliografia

[1] Pomoc on-line programu statistica, 2005. Dostępny w Internecie: http://www.statistica.pl/textbook/stcluan.html.

[2] Agrawal, R., Imieliński, T., and Swami, A. *Mining association rules between sets of items in large databases.* Proceedings of the 1993 ACM SIGMOD international conference on Management of data - SIGMOD '93, 1993.

[3] Agyemang, M., Barker, K., and Alhajj, R. A comprehensive survey of numeric and symbolic outlier mining techniques. *Intelligent Data Analysis* (2006).

[4] Alayon, S., Robertson, R., Warfield, S. K., and Ruiz-Alzola, J. Fuzzy system for helping medical diagnosis of malformations of cortical development. *Journal of Biomedical Informatics* (2007).

[5] Allison, P. D. *Missing Data (1st ed.).* Thousand Oaks: Sage Publications, 2001.

[6] Angiulli, F. Outlier mining in large high-dimensional data sets. *Knowledge and Data Engineering, IEEE Transactions on Knowledge and Data Engineering* (2005).

[7] Ankerst, M., Breunig, M. M., Kriegel, H. P., and Sander, J. Optics: Ordering points to identify the clustering structure. *ACM SIGMOD international conference on Management of data.* (1999).

[8] Bache, K., and Lichman, M. UCI machine learning repository, 2013.

[9] Baronti, E., Casini, E., and Papini, P. L. Centroids, centers, medians: What is the difference? *Geometriae Dedicata 68* (1997).

[10] Bass, T. *Rete, Rete II, Rete III & Rete in TIBCO BusinessEvents*. http://www.slideshare.net/TimBassCEP/ss-presentation-716373, 2006.

[11] Batory, D. *The LEAPS Algorithms*. http://reports-archive.adm.cs.cmu.edu/anon/1995/CMU-CS-95-113.pdf, 1995.

[12] Bazan, J., Nguyen, H. S., Nguyen, S. H., Synak, P., and Wróblewski, J. *Rough set algorithms in classification problem.* Rough Set Methods and Applications. Physica-Verlag, 2000.

[13] Bazan, J., and Szczuka, M. Rses and rseslib - a collection of tools for rough set computations. *Rough Sets and Current Trends in Computing* (2000).

[14] Bazan, J. G., Szczuka, M. S., Wojna, A., and Wojnarski, M. *On the Evolution of Rough Set Exploration*. Rough Sets and Current Trends in Computing. Lecture Notes in Artificial Inteligence, 2004.

[15] Ben-Gal, I. *Encyclopedia of Statistics in Quality & Reliability*. Wiley & Sons, 2007, ch. Bayesian Networks.

[16] Berkhin, P. *Survey of Clustering Data Mining Techniques*. Accrue Software, 2002.

[17] Berkhin, P. Survey of clustering data mining techniques. Tech. rep., 2002.

[18] Berry, M. J. A., and Linoff, G. *Data Mining Techniques for Marketing, Sales and Customer Support*. John Wiley & Sons, 1996.

[19] Bolc, L., and Coombs, M. J. *Expert System Applications*. Springer-Verlag, 1988.

[20] Carnap, R. Logical foundations of probability. *University of Chicago Press* (1950).

[21] Chandru, V., and Hooker, J. *Optimization methods for logical inference*. John Wiley and Sons, 1999.

[22] Chang, P. C., and Liao, T. W. Combining som and fuzzy rule base for flow time prediction in semiconductor manufacturing factory. *Applied Soft Computing* (2006).

[23] Chang, P. C., Liu, C. H., and Wang, Y. W. A hybrid model by clustering and evolving fuzzy rules for sales decision supports in printed circuit board industry. *Decision Support Systems* (2006).

[24] Chen, M. S., Chau, C. C., and Kabat, W. C. *Decision support systems: A rule-based approach.* Artificial Intelligence & Information Systems, 1985.

[25] Cholewa, W., and Pedrycz, W. *Systemy doradcze.* Skrypt Politechniki Śląskiej nr 1447, 1987.

[26] Chopra, R. *Artificial Intelligence: A Practical Approach.* S. Chand, 2012.

[27] Cormen, H., Leiserson, C. E., and Rivest, R. L. *Wprowadzenie do algorytmów.* Wydawnictwa Naukowo-Techniczne, 1990.

[28] Czogała, E., and Pedrycz, W. *Elementy i metody teorii zbiorów rozmytych.* Polskie Wydanictwa Naukowe, 1985.

[29] da Fontura Costa, L., and Jr, R. M. C. *Shape Analysis and Classification: Theory and Practice.* CRC Press, 2001.

[30] Deja, R. *Zastosowanie teorii zbiorów przybliżonych w analizie konfliktów. Rozprawa doktorska.* Instytut Podstaw Informatyki Polskiej Akademii Nauk, 2000.

[31] Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society* (1977).

[32] Digia. The qt project, 2013. http://qt-project.org/.

[33] Enders, C. K. *Applied Missing Data Analysis (1st ed.).* New York: Guildford Press, 2010.

[34] Ester, M., Kriegel, H. P., Sander, J., and Xiaowej, X. A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (1996).

[35] Fay, A. A fuzzy knowledge-based system for railway traffic control. *Engineering Applications of Artificial Intelligence* (2000).

[36] Feigenbaum, A., Nii, P., and McCorduck, P. *The Rise of the Expert Company.* Times Books, 1988.

[37] Flamant, B., and Girard, G. Intelligence service: construisez votre propre systeme expert = intelligence service: build your own expert system.

[38] Forgy, C. L. *On the efficient implementation of production systems.* PhD thesis, Carnegie-Mellon University, 1979.

[39] Forgy, C. L. Rete: A fast algorithm for the many pattern many object pattern match problem. *Artificial Intelligence* (1981).

[40] Gan, G., Ma, C., and Wu, J. *Data Clustering Theory: Algorithms and Applications*. American Statistical Association, 2007.

[41] Gaschnig, J., Klahr, P., Pople, H., Shortliffe, E., and Terry, A. Evaluation of expert systems: Issues and case studies. *Building expert systems 1* (1983), 241–278.

[42] Gatnar, E. *Symboliczne metody klasyfikacji danych.* PWN, 1998.

[43] Gevarter, W. B. The nature and evaluation of commercial expert system building tools. *Computer 20*, 5 (1987), 24–41.

[44] Gim, G., and Whalen, T. Logical second order models: Achieving synergy between computer power and human reason. *Information Sciences* (1999).

[45] Golec, A., and Kahya, E. fuzzy model for competency-based employee evaluation and selection. *A Computers & Industrial Engineering* (2007).

[46] Grogono, P., Batarekh, A., Preece, A., Shinghal, R., and Suen, C. Expert system evaluation techniques: a selected bibliography. *Expert Systems 8*, 4 (1991), 227–239.

[47] Grogono, P. D., Preece, A. D., Shingal, R., and Suen, C. Y. A review of expert systems evaluation techniques. Tech. rep., AAAI Technical Report, 1993.

[48] Grzegorczyk, A. *Zarys logiki matematycznej.* PWN, 1984.

[49] Grzymała-Busse, J., and Grzymała-Busse, W. An experimental comparison of three rough set approaches to missing attribute values. *Transactions on Rough Sets* (2007).

[50] Grzymała-Busse, J. W. *LERS – A Data Mining System.* The Data Mining and Knowledge Discovery Handbook. Springer, 2005.

[51] Grzymała-Busse, J. W. A closest fit approach to missing attribute values. *Recent Advances in Intelligent Information Systems* (2009).

[52] Guilan, K., Dong-Ling, X., and Jian-Bo, Y. Clinical decision support systems: A review on knowledge representation and inference under uncertainties. *International Journal of Computational Intelligence Systems* (2008).

[53] Guimaraes, A. C. F., and Lapa, C. M. F. Fuzzy inference system for evaluating and improving nuclear power plant operating performance. *Annals of Nuclear Energy* (2004).

[54] Guimaraes, A. C. F., and Lapa, C. M. F. Fuzzy inference to risk assessment on nuclear engineering systems. *Applied Soft Computing* (2007).

[55] Gustafson, D. E., and Kessel, W. C. Fuzzy clustering with a fuzzy covariance matrix,. *IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes* (1978).

[56] Halkidi, M., Batistakis, Y., and Vazirgiannis, M. *On Clustering Validation Techniques.* Journal of Intelligent Information Systems, 2001.

[57] Halkidi, M., Batistakis, Y., and Vazirgiannis, M. *Cluster validity methods: Part I.* ACM SIGMOD Record, 2002.

[58] Halkidi, M., Batistakis, Y., and Vazirgiannis, M. *Clustering Validity Checking Methods: Part II.* ACM SIGMOD Record, 2002.

[59] Hanson, E., and Hasan, M. S. Gator: An optimized discrimination network for active database rule condition testing. Tech. rep., 1993.

[60] Heckerman, D. The certainty-factor model. *https://research.microsoft.com/en-us/um/people/heckerman/h92encyclopedia.pdf* (1992).

[61] Heckerman, D. An empirical comparison of three inference methods. *CoRR abs/1304.2357* (2013).

[62] Heckerman, D., and Horovitz, E. The myth of modularity in rule-based systems. *Proceedings of the Second Workshop on Uncertainty in Artificial Intelligence* (1986).

[63] Heckerman, D. E. *Probabilistic Similarity Networks.* The MIT Press, Cambridge, Massachusetts, London, England, 1991.

[64] Hopgood, A. A. *Intelligent Systems for Engineers and Scientists, Third Edition.* CRC Press, Taylor & Francis Group, 2012.

[65] Hotteling, H. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology* (1933).

[66] Ignizio, J. P. *An introduction to Expert Systems.* McGraw-Hill, 1991.

[67] Jach, T. *Grupowanie jako metoda eksploracji wiedzy w systemach wspomagania decyzji. Analiza algorytmów hierarchicznych.* Praca licencjacka. Uniwersytet Śląski., 2008.

[68] Jach, T. Analiza gridowych algorytmów grupowania. Master's thesis, Uniwersytet Śląski, 2010.

[69] Jach, T. Gridowe algorytmy grupowania w grupowaniu danych tekstowych. *Systemy wspomagania decyzji. Wydawnictwo Uniwersytetu Śląskiego* (2010).

[70] Jach, T. Wnioskowanie w systemach z wiedzą niepełną. *Systemy wspomagania decyzji. Wydawnictwo Uniwersytetu Śląskiego* (2011).

[71] Jach, T. Wybrane aspekty wnioskowania w systemach z wiedzą niepełną. *Systemy wspomagania decyzji. Wydawnictwo Uniwersytetu Śląskiego* (2012).

[72] Jach, T. Metody wyznaczania współczynnika niepełności wiedzy w systemach z wiedzą niepełną. *Systemy wspomagania decyzji. Wydawnictwo Uniwersytetu Śląskiego* (2013).

[73] Jackson, P. *Introduction to Expert Systems.* Addison Wesley, 1999.

[74] Jelonek, J., Krawiec, K., Słowiński, R., Stefanowski, J., and Szymas, J. *Neural networks rough sets – Comparison combination for classification of histological pictures.* Rough Sets, Fuzzy Sets and Knowledge Discovery (RSKD'93. Springer–Verlag & British Computer Society, 1994.

[75] Joshi, M. *An Extensive Survey of Clustering Methods for Data Mining.* University of Minnesota, 2002.

[76] Kaufman, L., and Rousseeuw, P. J. *Finding Groups in Data: An Introduction to Cluster Analysis.* Wiley, 1990.

[77] Klir, G. J., and Yuan, B. *Fuzzy sets and fuzzy logic.* Prentice Hall New Jersey, 1995.

[78] Kolatch, E. *Clustering Algorithms for Spatial Databases: A Survey.* University of Maryland, 2001.

[79] Komorowski, J., Pawlak, Z., Polkowski, L., and Skowron, A. *Rough sets: A tutorial.* Rough–Fuzzy Hybridization: A New Trend in Decision–Making. Springer–Verlag, 1999.

[80] Korbicz, J., Obuchowicz, A., and Uciński, D. *Sztuczne sieci neuronowe: podstawy i zastosowania.* Akademicka Oficyna Wydawnicza, 1994.

[81] Koronacki, J., and Mielniczuk, J. *Statystyka dla studentów kierunków technicznych i przyrodniczych.* Wydawnictwa Naukowo Techniczne, 2006.

[82] Kovacs, F., Legany, C., and Babos, A. Cluster validity measurement techniques. *Proceedings of the 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases.* (2006).

[83] Kryszkiewicz, M. Rules in incomplete systems. *Information Sciences* (1999).

[84] Kumar, R., Novak, J., and Tomkins, A. *Structure and Evolution of Online Social Networks.* Link Mining: Models, Algorithms, and Applications, 2010.

[85] Kuter, U., and Golbeck, J. Sunny: A new algorithm for trust inference in social networks using probabilistic confidence models. *Association for the advancement of Artificial Intelligence* (2007).

[86] Lauritzen, S. L., and Spiegelhalter, D. J. Local computations with probabilities on graphical structures and their application to expert systems (with discussion). *Journal of the Royal Statistical Society* (1988).

[87] Leśniewski, S. *Collected Works.* Kluwer, 1992.

[88] Leondes, C. T. *Expert systems: the technology of knowledge management and decision making for the 21st century.* Academic Press, 2002.

[89] Liu, Y., and Phinn, S. R. Modelling urban development with cellular automata incorporating fuzzy-set approaches, computers. *Environment and Urban Systems* (2003).

[90] Maedche, A., and Staab, S. *Ontology learning for the semantic web.* Intelligent Systems, IEEE, 2001.

[91] Mahoney, J. J., and Mooney, R. J. Comparing methods for refining certainty factor rule-bases. In *Proceedings of the Eleventh International Workshop on Machine Learning (ML-94)* (1994).

[92] Mercer, D. P. Clustering large datasets. *Linacre College* (2003). Dostępny w Internecie: `http://www.stats.ox.ac.uk/~mercer/`.

[93] Mercer, D. P. Clustering large datasets. *Linacre College* (2003).

[94] Michalik, K. *Sphinx 2.2 – dokumentacja pakietu.* AITECH, Katowice, 1998.

[95] Miranda, P., Isaias, P., and Crisostomo, M. *Human Interface and the Management of Information. Methods, Techniques and Tools in Information Design.* Springer Berlin Heidelberg, 2007, ch. Expert Systems Evaluation Proposal.

[96] Miranker, D. P. Treat: A better match algorithm for ai production systems. Tech. rep., Department of Computer Sciences, University of Texas at Austin, 1987.

[97] Mulawka, J. *Systemy Ekspertowe.* WNT, 1996.

[98] Myatt, G. *Making Sense of Data a Practical Guide to Exploratory Data Analysis and Data Mining.* John Wiley and Sons, 2007.

[99] Negnevitsk, M. Artificial intelligence: A guide to intelligent systems. *QPearson Education Limited* (2002).

[100] Nguyen, H. T., and Sugeno, M., Eds. *Fuzzy Systems: Modeling and Control.* Kluwer Academic Publishers, 1998.

[101] Nguyen, S. H., Skowron, A., Synak, P., and Wróblewski, J. *Knowledge Discovery in Databases: Rough Set Approach.* Boston: Kluwer Academic Publishers, 1991.

[102] Niederliński, A. *Regułowe systemy ekspertowe.* Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, 2000.

[103] Nikolopulos, C. *Expert Systems.* Marcel Dekker, 1997.

[104] Nilsson, U., and Maluszyński, J. *Logic, Programming and Prolog (2ed).* John Wiley & Sons Ltd., 1995.

[105] Norris, J. R. Markov chains. *Cambridge University Press* (1998).

[106] Nowak-Brzezińska, A. *Złożone bazy wiedzy: struktura i procesy wnioskowania. Rozprawa doktorska.* Uniwersytet Śląski, 2009.

[107] Nowak-Brzezińska, A., and Jach, T. Wnioskowanie w systemach z wiedzą niepełną. *Studia Informatica, Zeszyty Naukowe Politechniki Śląskiej* (2011).

[108] Nowak-Brzezińska, A., and Jach, T. Wybrane aspekty wnioskowania w systemach z wiedzą niepełną. *Studia Informatica, Zeszyty Naukowe Politechniki Śląskiej* (2012).

[109] Nowak-Brzezińska, A., and Jach, T. Metoda współczynników niepełności wiedzy w systemach wspomagania decyzji. *Studia Informatica, Zeszyty Naukowe Politechniki Śląskiej* (2013).

[110] Nowak-Brzezińska, A., Jach, T., and Xięski, T. Analiza hierarchicznych i niehierarchicznych algorytmów grupowania dla dokumentów tekstowych. *Studia Informatica, Zeszyty Naukowe Politechniki Śląskiej* (2009).

[111] Nowak-Brzezińska, A., Jach, T., and Xięski, T. Finding a relevant document in the clusters of documents' characteristics. *Intelligent Information Systems 2010, Publishing House of University of Podlasie* (2010).

[112] Nowak-Brzezińska, A., Jach, T., and Xięski, T. Wybór algorytmu grupowania a efektywność wyszukiwania dokumentów. *Studia Informatica, Zeszyty Naukowe Politechniki Śląskiej* (2010).

[113] Nowak-Brzezińska, A., and Wakulicz-Deja, A. Kryteria stopu algorytmu grupowania reguł w hierarchicznych bazach wiedzy. *Systemy Wspomagania Decyzji* (2006).

[114] Pankowski, T. *Integracja i eksploracja danych.* Wymiana Informacji i Interaktywne Komunikowanie Medialne, 2003.

[115] Paszek, P., and Marszał-Paszek, B. *Deterministic and nondeterministic decision rules in classification process.* Medical knowledge bases and management. Journal of medical informatics & technologies, 2010.

[116] Pawlak, Z. Rough sets. *International Journal of Parallel Programming* (1982).

[117] Pawlak, Z. *System informacyjne - podstawy teoretyczne.* Wydawnictwa Naukowo Techniczne, 1983.

[118] Pawlak, Z. *O Konfliktach.* Państwowe Wydawnictwo Naukowe, 1987.

[119] Pawlak, Z. *Rough Sets: Theoretical aspects of reasoning about data.* Seventh International Fuzzy Systems Association World Congress (IFSA'1997). Academia, Prague, 1997.

[120] Pawlak, Z., and Skowron, A. A rough set approach to decision rules generations. *Proceedings of the Workshop W12: The Management of Uncertainty in AI at 13th IJCAI* (1993).

[121] Pearl, J. Probabilistic reasoning in intelligent systems: Networks of plausible inference. *Morgan Kaufmann* (1988).

[122] Pearl, J. *Probabilistic Reasoning in Intelligent Systems. Networks of Plausible Inference. Representation and Reasoning Series (2nd printing ed.).* Morgan Kauffman, 1988.

[123] Pearl, J., and Russel, S. *Handbook of Brain Theory and Neural Networks.* MIT Press, 2000, ch. Bayesian networks. Report.

[124] Pereira, R., Sanchez, J. L., and Rives, J. Knowledge-based maneuver and fire support planning. *Expert Systems with Applications* (1999).

[125] Pulatova, S. *Covering (Rule–Based) Algorithms.* Lecture Notes in Data Mining. World Scientific, Singapore, 2006.

[126] Raś, Z., and Joshi, S. Query answering system for an incomplete dkbs. *Fundamenta Informaticae* (1997).

[127] Reichgelt, H. *Knowledge Representation : An AI Perspective.* Ablex Publishing Corporation, 1991.

[128] Riesbeck, C. K., and Schank, R. C. *Inside Case-Based Reasoning.* Lawrence Erlbaum Associates, 1989.

[129] Rokach, L., and O, M. *Data mining with decision trees: theory and applications.* World Scientific, 2008.

[130] Rothenberg, J., Paul, J., Kameny, I., Kipps, J. R., and Swenson, M. Evaluation expert system tools. a framework and methodology. Tech. rep., Defense Advanced Research Projects Agency.

[131] Rubin, D. B. Inference and missing data. *Biometrika 63* (1976).

[132] Rubin, D. B., and Little, R. J. A. *Statistical analysis with missing data (2nd ed.).* New York: Wiley, 2002.

[133] Russel, S., and Norvig, P. *Artificial Intelligence: A Modern Approach (3rd Edition).* Prentice Hall Series in Artificial Intelligence, 2010.

[134] Salim, M. D., Villavicencio, A., and Timmerman, M. A method for evaluating expert system shells for classroom instruction. *Journal of Industrial Technology* (2002).

[135] Salton, G. Automatic information organization and retreival. *McGraw-Hill* (1975).

[136] Schnupp, P., and Huu, C. T. N. Expertensystem-praktikum. *Springer-Verlag* (1987).

[137] Shafer, G. *A mathematical theory of evidence.* Princeton, 1976.

[138] Sheskin, D. *Handbook of Parametric and Nonparametric Statistical Procedures.* CRC Press, 2004.

[139] Shortliffe, E. H., and Buchanan, B. G. A model of inexact reasoning in medicine. *Mathematical Biosciences* (1975).

[140] Shortliffe, E. H., and Buchanan, B. G., Eds. *Rule-Based Expert Systems. The MYCIN Experiments of the Stanford Heuristic Programming Project.* Addison-Wesley Publishing Company, 1984, ch. EMYCIN: A knowledge Engineer's Tool for Constructing Rule-Based Expert Systems.

[141] Sikorski, R. *Boolean Algebras.* Springer-Verlag Berlin, 1969.

[142] Simiński, R., Nowak-Brzezińska, A., Jach, T., and Xięski, T. *Towards a Practical Approach to Discover Internal Dependencies in Rule-Based Knowledge Bases.* 6th International Conference, Rough Sets and Knowledge Technology, 2011.

[143] Skowron, A., and Rauszer, C. *Intelligent Decision Support - Handbook of Applications and Advances in Rough Set Theory.* Kluwer Academic Publishers, 1992, ch. The discernibility matrices and functions in information systems.

[144] Slatter, P. E. *Building expert systems: cognitive emulation.* Halsted Press, 1987.

[145] Sneath, P. H., and Sokal, R. R. *Numerical Taxonomy.* W.H. Freeman, 1973.

[146] Sonmez, H., Gokceoglu, C., and Ulusay, R. A mamdani fuzzy inference system for the geological strength index (gsi) and its use in slope stability assessments. *International Journal of Rock Mechanics and Mining Sciences* (2004).

[147] Sorgaad, P. Evaluating expert system prototypes. *AI & SOCIETY* (1991).

[148] Stefanowski, J. Algorytmy indukcji reguł decyzyjnych w odkrywaniu wiedzy. rozprawa habilitacyjna. Seria Rozprawy nr 361.

[149] Stefanowski, J., and Tsoukitas, A. Incomplete information tables and rough classification. *Computational Intelligence* (2001).

[150] Stein, B., Meyerzu-Eissen, S., and Wissbrock, F. On cluster validity and the information need of users. *3rd IASTED Int. Conference on Artificial Intelligence and Applications (AIA 03)* (2003).

[151] Stolfo, S. J., and Miranker, D. P. Dado:a parallel processor for expert systems. In *Proceedings International Conference on Parallel Processing* (1984).

[152] Swinburne, R. G. An introduction to confirmation theory. *Methuen* (1973).

[153] Sworowska, A. Proces konstruowania map przepływów wiedzy. *Grant NCN nr 4160/B/H03/2011/40: Model racjonalizacji procesów innowacyjnych we wdrażaniu strategii rozwoju regionu* (2012).

[154] Tan, P. N., Steinbach, M., and Kumar, V. *Introduction to data mining.* Addison-Wesley, 2006.

[155] Theodoridis, S., and Koutroumbas, K. *Patern Recognition.* Academic Press, 1999.

[156] Łukasiewicz, J. *O zasadzie sprzeczności u Arystotelesa.* Państwowe Wydawnictwo Naukowe, 1987 (wznowienie).

[157] Ustundag, A., Kilinc, M. S., and Cevikcan, E. Fuzzy rule-based system for the economic analysis of rfid investments. *Expert Systems with Applications* (2010).

[158] Wakulicz-Deja, A. Systemy wspomagania decyzji. Wykład wygłoszony w Instytucie Informatyki.

[159] Wakulicz-Deja, A. *Podstawy systemów wyszukiwania informacji. Analiza metod.* Akademicka Oficyna Wydawnicza PLJ, 1995.

[160] Wakulicz-Deja, A., Nowak-Brzezińska, A., and Jach, T. Inference processes in decision support systems with incomplete knowledge. *Rough Sets and Knowledge Technology, Lecture Notes in Computer Science* (2011).

[161] Wakulicz-Deja, A., Nowak-Brzezińska, A., and Jach, T. *Inference Processes in Decision Support Systems with Incomplete Knowledge.* 6th International Conference, Rough Sets and Knowledge Technology, 2011.

[162] Wakulicz-Deja, A., Nowak-Brzezińska, A., and Jach, T. Inference processes using incomplete knowledge in decision support systems - chosen aspects. *Rough Sets and Current Trends in Computing, Lecture Notes in Computer Science* (2012).

[163] Wakulicz-Deja, A., Nowak-Brzezińska, A., and Simiński, R. *Sztuczna Inteligencja - systemy ekspertowe.* Instytut Informatyki UŚl., wydanie elektroniczne, 2009.

[164] Wang, Y. W., and Hanson, E. A performance comparison of the rete and treat algorithms for testing database rule conditions. *Eighth International Conference on Data Engineering* (1992).

[165] Weiner, P. Linear pattern matching algorithms. In *Switching and Automata Theory, 1973. SWAT '08. IEEE Conference Record of 14th Annual Symposium on* (1973), pp. 1–11.

[166] Wentworth, J. A. *Verification, Validation and Evaluation of Expert Systems.* FHWA Handbook, 1995.

[167] Wierzchoń, S. T., and Kłopotek, M. A. *Evidential Reasoning. An Interpretative Investigation.* Wydawnictwo Akademii Podlaskiej, 2002.

[168] Witten, I. H., and Frank, E. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition.* Morgan Kaufmann Series in Data Management Systems, 2005.

[169] Yatskiv, I., and Gusarova, L. The methods of cluster analysis results validation. *Proceedings of International Conference RelStat'04.* (2004).

[170] Zadeh, L. A. Fuzzy sets. *Information and Control 8* (1965).

Spis rysunków

Spis tabel

Spis algorytmów