

POLITECHNIKA ŚLĄSKA  
WYDZIAŁ AUTOMATYKI, ELEKTRONIKI I INFORMATYKI

ROZPRAWA DOKTORSKA

# **Indukcja reguł akcji na podstawie metody sekwencyjnego pokrywania**

**Paweł Matyszok**

Promotor: dr hab. Marek Sikora, prof. Pol. Śl.

Promotor pomocniczy: dr inż. Łukasz Wróbel

Gliwice 2022



# Spis treści

<b>1</b>	<b>Wprowadzenie</b>	<b>3</b>
1.1	Teza pracy . . . . .	8
1.2	Cele pracy . . . . .	8
1.3	Układ pracy . . . . .	9
<b>2</b>	<b>Podstawowe definicje i metody indukcji reguł klasyfikacyjnych</b>	<b>11</b>
2.1	Sposób reprezentacji reguł . . . . .	11
2.2	Sposób oceny reguł klasyfikacyjnych. Klasyfikacja za pomocą reguł klasyfikacyjnych . . . . .	13
2.2.1	Miary oceny jakości reguł klasyfikacyjnych . . . . .	14
2.2.2	Sposób klasyfikacji regułami decyzyjnymi . . . . .	19
2.3	Wybrane algorytmy indukcji reguł klasyfikacyjnych . . . . .	22
2.3.1	Pokryciowy algorytm indukcji reguł decyzyjnych . . . . .	25
2.3.2	Indukcja reguł decyzyjnych na podstawie strategii przeszuki- wania wyczerpującego . . . . .	30
2.4	Reguły akcji . . . . .	36
2.5	Wybrane algorytmy indukcji reguł akcji . . . . .	42
2.5.1	Indukcja reguł akcji wprost z tablicy decyzyjnej . . . . .	43
2.5.2	Indukcja reguł akcji na podstawie zbiorów reguł decyzyjnych . . . . .	46
2.5.3	Generowanie reguł akcji z zastosowaniem strategii wyczerpującej . . . . .	47
2.5.4	Odkrywanie reguł akcji na podstawie drzew decyzyjnych . . . . .	49
2.5.5	Równoległe odkrywanie reguł akcji . . . . .	51
2.5.6	Zastosowanie reguł akcji . . . . .	51
<b>3</b>	<b>Indukcja reguł akcji</b>	<b>54</b>
3.1	Indukcja reguł akcji z zastosowaniem paradygmatu sekwencyjnego pokrywania . . . . .	55
3.2	Wpływ kierunku indukcji na postać reguł akcji . . . . .	63

3.3	Inne rodzaje reguł akcji . . . . .	68
3.3.1	Reguły akcji w zadaniu regresji . . . . .	68
3.3.2	Reguły akcji w zadaniu analizy przeżycia . . . . .	73
<b>4</b>	<b>Rekomendacje na podstawie zbiorów reguł akcji. Weryfikacja reguł akcji i rekomendacji.</b>	<b>75</b>
4.1	Zastosowanie reguł akcji i rekomendacji . . . . .	77
4.2	Meta-reguły . . . . .	80
4.2.1	Meta-tablica . . . . .	81
4.2.2	Indukcja rekomendacji . . . . .	83
4.3	Weryfikacja reguł akcji i rekomendacji . . . . .	87
4.3.1	Uczenie modelu akcji. Ważność klasyfikatora weryfikującego. .	91
4.3.2	Transformacja przykładów. Symulowanie zastosowania akcji. .	93
4.3.3	Klasyfikacja i testowanie klasyfikatora. Metryki modelu akcji.	96
4.3.4	Rekomendacje a problemy regresyjne . . . . .	98
<b>5</b>	<b>Eksperymenty</b>	<b>103</b>
5.1	Parametry algorytmów . . . . .	103
5.2	Testy statystyczne w ocenie jakości algorytmów . . . . .	107
5.3	Reguły akcji w zadaniu analizy danych klasyfikacyjnych . . . . .	112
5.3.1	Charakterystyka zbiorów danych . . . . .	112
5.3.2	Indukcja reguł „w przód” . . . . .	114
5.3.3	Indukcja reguł „wstecz” . . . . .	122
5.3.4	Fuzja wariantów „w przód” i „wstecz” . . . . .	129
5.3.5	Wybór najlepszego wariantu . . . . .	133
5.4	Reguły akcji w zadaniu analizy danych regresyjnych . . . . .	144
5.4.1	Charakterystyka zbiorów danych . . . . .	144
5.4.2	Indukcja reguł akcji i rekomendacji z danych regresyjnych . .	147
<b>6</b>	<b>Podsumowanie</b>	<b>156</b>
6.1	Kierunki rozwoju . . . . .	157

# 1 Wprowadzenie

W ostatnich dziesięcioleciach obserwujemy, jak komputery wdrażane są do niemal każdej dziedziny ludzkiego życia. Malejące koszty przestrzeni dyskowej i rozpowszechnienie komputerów pozwalają na gromadzenie danych opisujących różne aspekty działalności człowieka, których w przeszłości, ze względu na ograniczenia technologiczne, nie można było zachować. „*Dane to nowa ropa*” (ang. *Data is the new oil*) to powiedzenie ukute przez brytyjskiego matematyka Clive’a Humby’ego, wskazujące na fakt, że w XXI wieku dane są nabytkiem równie cennym, co ropa naftowa w wiekach ubiegłych. Jednak dane, podobnie jak ropa, muszą zostać przetworzone by zaowocować korzyściami dla ich właścicieli. Ogrom i dostępność danych prowadzą do gwałtownego rozwoju metod ich przetwarzania i eksploracji prowadzącej do odkrywania użytecznej wiedzy.

Eksploracja danych jest procesem wieloetapowym, wymagającym zaangażowania zarówno eksperta dziedzinowego jak i specjalisty w zakresie metod analizy. Rosnące znaczenie zastosowania metod analizy danych doprowadziło do wytworzenia branżowych standardów prowadzenia i wdrażania projektów, jak CRISP-DM [79] czy CRISP-DM(Q) [90] będący jego rozszerzeniem ze szczególnym uwzględnieniem wdrażania metod wykorzystujących uczenie maszynowe. Niezależnie od stosowanego standardu, w procesie eksploracji danych znajdują się etapy takie jak: określenie celu analizy, zbadanie przydatności dostępnych danych, przetworzenie i oczyszczenie danych, przeprowadzenie modelowania czyli właściwej analizy, ewaluacja modelu i ostateczne wdrożenie go do procesów biznesowych. Wdrożony model powinien być także monitorowany pod kątem jego jakości w zetknięciu z nowymi danymi.

Etap modelowania często opiera się o zastosowanie metod uczenia maszynowego [105] takich jak grupowanie danych (kNN, K-medoids), maszyny wektorów podpierających (SVM), indukcja reguł i drzew czy wnioskowanie Bayesowskie.

Uczenie maszynowe określa szeroką gamę algorytmów, które są zdolne do poprawiania wyników swego działania na podstawie podawanych na ich wejście danych. Interesującą definicję uczenia maszynowego podał w 1997 roku amerykański informa-

tyk Tom Mitchell: „Program komputerowy nazwiemy uczącym się z doświadczenia  $E$  w odniesieniu do klasy zadań  $T$  i miary jakości  $P$ , jeśli jego jakość w zadaniach  $T$ , mierzona przez  $P$ , poprawia się wraz z doświadczeniem  $E$ ” [62]. Systemem uczącym się jest więc taki system, który zmienia swoje wewnętrzne parametry, by rozpoznać i opisać dane, i na tej podstawie rozwiązywać pewien problem. Systemy uczenia maszynowego znajdują zastosowanie wszędzie tam, gdzie podanie analitycznego rozwiązania jest trudne lub niemożliwe, jak rozpoznawanie obrazów, mowy, filtrowanie poczty elektronicznej czy predykcja przyszłych wartości ciągów czasowych.

Metody uczenia maszynowego dzieli się na trzy główne kategorie: uczenie nadzorowane, nienadzorowane i uczenie ze wzmacnianiem, czasami w tę nomenklaturę włącza się także metody redukcji wymiarowości danych. W uczeniu nadzorowanym dane uczące zawierają zarówno dane wejściowe jak i wyjściowe, na podstawie których algorytm jest w stanie nauczyć się jak wejścia wpływają na wyjścia, by w przyszłości wykorzystać tę wiedzę do przewidywania wartości wyjść dla pewnego zestawu wejść. W uczeniu nienadzorowanym nieznane są wartości wyjść, a program grupuje dane w klastry na podstawie odkrytego podobieństwa opisywanych obiektów. W uczeniu ze wzmacnianiem rolę danych uczących pełni przygotowane środowisko, z którym oddziałuje agent (program). Każda akcja agenta związana jest z odpowiedzią środowiska, które przydziela pewną nagrodę. Zadaniem programu jest poznanie środowiska i takie dostosowania swoich polityk działania, by zmaksymalizować otrzymywaną nagrodę.

Niezależnie od wybranej metody uczenia maszynowego, otrzymuje się reprezentację wiedzy o analizowanych danych. Reprezentacja ta może być trudna do zrozumienia, jak w przypadku wielowarstwowych perceptronów. Łatwą i intuicyjną formą reprezentacji wiedzy są drzewa i reguły. Z tego powodu indukcja drzew i reguł jest najczęściej stosowana w zadaniach, w których niezwykle ważne jest zrozumienie i czytelność modelu opisującego dane. Nie zważając na cechę czytelności uzyskanego modelu, można by stwierdzić, że maszyny wektorów podpierających czy sztuczne sieci neuronowe (w szczególności tzw. sieci głębokie [36]) przewyższają zdolności innych metod w zadaniach regresji i klasyfikacji, jednak w wielu zastosowaniach istotne jest to, w jaki sposób podjęto pewną decyzję, a nie tylko sama decyzja. Z tego też powodu prowadzi się pracę nad interpretowalnością złożonych modeli [41] (ang. *Explainable Artificial Intelligence, XAI*). Wyjaśnianie działania modeli jest szczególnie istotne przy zastosowaniu metod uczenia głębokiego w dziedzinach takich jak rozpoznawanie obrazów, szczególnie medycznych, czy przetwarzaniu języka naturalnego. Jedną z

popularnych metod pozwalających wyjaśnić działanie dowolnego klasyfikatora jest LIME [77] (ang. *Local Interpretable Model-agnostic Explanations*). Dzięki metodzie wyjaśniającej działanie klasyfikatora, jesteśmy w stanie stwierdzić, przykładowo, że analizując zdjęcia rentgenowskie płuc algorytm nauczył się wykrywać zmiany chorobowe, a nie specyficzny układ kości widocznych na zdjęciu, będący wynikiem pozycji przyjętej przed aparatem przez zmęczonego chorobą pacjenta. W odwodzie zawsze pozostają metody charakteryzujące się czytelnością modelu i wystarczająco wysoką skutecznością jak drzewa i reguły.

Reguły logiczne formalizują odkrytą wiedzę w postaci wyrażeń warunkowych opisujących lokalne zależności między wartościami atrybutów w pewnym zbiorze danych. Wśród wszystkich atrybutów (cech) wyróżnia się jeden atrybut zwany decyzyjnym i oznaczający tę cechę opisywanych obiektów, która jest interesująca ze względu na cel prowadzonego modelowania. Reguły znajdują swoje zastosowanie w zadaniach klasyfikacji (przypisywania obiektów do właściwej dla nich klasy), asocjacji (odkrywanie zestawów cech często występujących razem), regresji (przewidywania wartości liczbowej atrybutu decyzyjnego) i analizie przeżycia (przewidywanie czasu, jaki upłynie do zajścia pewnego zdarzenia) [104].

Innym, stosunkowo nowym obszarem zastosowań reguł, jest badanie zależności prowadzących do zmiany wartości atrybutu decyzyjnego. Narzędziem umożliwiającym opis zbioru danych pod tym kątem są *reguły akcji*. Reguły akcji zostały po raz pierwszy opisane przez Rasia i Wieczorkowską w roku 2000 [75]. Reguła akcji w swej budowie podobna jest do reguły decyzyjnej (rozdział 2.1 i 2.4), przy czym zamiast wiązać atrybut decyzyjny z jedną wartością, wskazuje na wartość źródłową i docelową. Podobna zmiana zachodzi w części warunkowej reguły. Zmienia się także znaczenie reguły. O ile regułę decyzyjną należy rozumieć jako stwierdzenie „Jeżeli spełnione są pewne warunki dotyczące cech obiektu, to zachodzi podana decyzja”, to reguła akcji odpowiada na pytanie: „Jak muszą zmienić się cechy obiektu, by zaszła zmiana w decyzji?”.

Reguły akcji mogą stanowić narzędzie nie tylko opisujące zależności między atrybutami warunkowymi a atrybutem decyzyjnym, ale mogą też pełnić rolę bardzo praktycznej rekomendacji, ponieważ wdrażając pewne operacje, jesteśmy w stanie zmienić stan obiektu zgodnie z sugestiami podanymi w regule akcji i oczekiwać odpowiedniej zmiany wartości atrybutu decyzyjnego. Proces odkrywania fizycznych działań prowadzących do pożądaných zmian w wartościach atrybutów nazywa się

odkrywaniem meta-akcji [73]. W tej pracy nie zajmujemy się jednak meta-akcjami, lecz odkrywaniem reguł akcji.

Odkrywanie reguł akcji jest częścią szerszej koncepcji odkrywania akcji (ang. *Action Mining*, AM) zwanej także odkrywaniem użytecznej wiedzy (ang. *Actionable Knowledge*). Większość metod analizy danych pozwala na odkrywanie pewnych wzorców, jednak nie udziela odpowiedzi na pytanie, jakie działania należy podjąć, aby uzyskać korzyść [43], a przecież ostatecznie uzyskanie pewnej korzyści jest celem każdego procesu eksploracji danych. Odkryte wzorce w danych są raczej środkiem, niż celem. Odkrywanie akcji jest dziedziną zajmującą się rozwojem metod, które potrafią wskazać, w jaki sposób wpływać na dane, czy raczej obiekty przez nie reprezentowane, aby osiągnąć założone cele.

W ramach odkrywania akcji wyróżnia się generalnie dwa podejścia: luźno powiązane (ang. *loosely coupled*) i ściśle powiązane (ang. *tightly coupled*) [19, 43]. W metodach luźno powiązanych, przeprowadza się analizę danych, i na podstawie uzyskanych wyników np. klasyfikacji próbuje się podać potencjalne akcje. W metodach z rodziny ściśle powiązanych, podawanie akcji jest zintegrowanym elementem procesu analizy danych, a akcje są wynikiem działania metody i obiektem jej optymalizacji.

Wśród metod odkrywania akcji możemy wyróżnić dalej dwie kategorie [97]. Pierwsza, stanowiąca etap odkrywania wiedzy, pozwala na odkrywanie akcji (w tym: reguł akcji). Druga, pozwala na odnalezienie najlepszych akcji dla zadanego przykładu, mających pozwolić na zmianę jego przypisania z klasy niepożądaney, do pożądaney. W niniejszej pracy omówione zostaną metody należące do obydwu kategorii.

Użytkownik metod odkrywania akcji może patrzeć na nie pod kątem dwóch aspektów: ich zdolności opisowych i prognostycznych. Przez zdolności opisowe będziemy rozumieć w jaki sposób akcje ujmują wiedzę o możliwych zmianach atrybutów i ich wpływie na wyróżniony atrybut decyzyjny. Zdolności prognostyczne określać będą przydatność odkrytych akcji pod kątem tego, jaki efekt przynosi ich wdrożenie w rzeczywistości, czy faktycznie postępowanie zgodnie z wiedzą ujętą w odkrytych akcjach przyniesie zamierzony skutek.

Indukcja reguł akcji zazwyczaj odbywa się na podstawie analizy danych przedstawionych w postaci systemu informacyjnego [44, 75, 76], a właściwie tablicy decyzyjnej definiowanej podobnie jak u Pawlaka [66]. W metodach tych zazwyczaj pośrednio konstruuje się reguły decyzyjne, a następnie na podstawie par reguł o sprzecznych decyzjach tworzy się reguły akcji [46, 75]. Rzadziej spotykane są metody odkrywania reguł akcji polegające na transformacji drzew decyzyjnych do postaci reguł akcji [107].



W pracy [96] wykorzystano teorię grafów do odkrywania reguł akcji i rekomendacji na podstawie istniejących już zbiorów reguł akcji decyzyjnych. Większość istniejących algorytmów potrafi odkrywać reguły akcji z danych klasyfikacyjnych, nierzadko metody te obarczone są innymi ograniczeniami takimi jak konieczność wcześniejszej dyskretyzacji danych czy wymóg braku wartości brakujących w analizowanym zbiorze danych.

Reguły akcji koncepcyjnie czerpią wiele z reguł decyzyjnych. Również metody ich odkrywania często są inspirowane metodami odkrywania klasycznych reguł decyzyjnych. Do tej pory jednak nie wykorzystano do indukcji reguł decyzyjnych paradygmatu sekwencyjnego pokrywania, który jest jednym z popularniejszych i efektywniejszych sposobów odkrywania reguł [31, 32]. Paradygmat sekwencyjnego pokrywania pozwala także na kierowanie indukcji reguł przy użyciu różnych kryteriów jakości, określających przydatność reguł w danym zastosowaniu, co dodatkowo zwiększa jego atrakcyjność przez możliwość łatwej modyfikacji metody do potrzeb jej użytkownika. Dodatkowo, metody stosujące podejście sekwencyjnego pokrywania w indukcji reguł decyzyjnych potrafią pracować na danych niekompletnych oraz atrybutach ciągłych. W literaturze rzadko poruszany jest także temat wdrażania reguł akcji w życie i oceny ich skuteczności. Oczywiście, pełną informację na temat skuteczności reguł akcji można uzyskać tylko w wyniku manipulacji rzeczywistych obiektów zgodnie z wiedzą zapisaną w regułach i śledzenia dalszych losów tak zmienionych obiektów celem późniejszej obserwacji, czy doszło do zmian cechy decyzyjnej. Proces ten może być kosztowny i czasochłonny, dlatego pożądana byłaby metoda modelowania cyfrowo tego procesu i zbadania potencjalnej skuteczności reguł jeszcze przed ich wdrożeniem w życie. Innym otwartym problemem jest wybór reguły akcji, która najlepiej nadawałaby się do wdrożenia. Problem ten wynika z faktu, że metody indukcji reguł akcji mogą prowadzić do takich zbiorów reguł, wśród których znajdzie się więcej niż jedna reguła, która pasowałaby do analizowanego przykładu. Innymi słowy, należy rozwiązać konflikt między regułami.

W dziedzinie odkrywania reguł decyzyjnych, paradygmat sekwencyjnego pokrywania znajduje zastosowanie w uznanych za skuteczne algorytmach [31, 32, 104]. Rodzi to nadzieje, że zastosowanie tego podejścia w dziedzinie reguł akcji pozwoli na stworzenie metody odkrywania akcji w ramach dobrze zbadanego podejścia, cieszącej się wszystkimi jego zaletami: możliwością sterowania postacią uzyskanych reguł, pracą na niekompletnych danych oraz atrybutach ciągłych. Dodatkowo paradygmat ten został też z powodzeniem zastosowany w odkrywaniu reguł regresyjnych. Jego

dostosowanie do potrzeb indukcji reguł akcji z danych regresyjnych pozwoliłoby zapłacić istniejącą lukę - wedle najlepszej wiedzy autora, do tej pory nie podano algorytmu indukcji reguł akcji z danych regresyjnych.

## 1.1 Teza pracy

**Zastosowanie w odkrywaniu reguł akcji paradygmatu sekwencyjnego pokrywania i odpowiednio dobranych kryteriów sterowania algorytmem indukcji reguł pozwala na uzyskanie modeli o dobrych zdolnościach prognozy stycznych i objaśniających.**

### Tezy pomocnicze

- Metodyka wymieniona w głównej tezie pracy może być zastosowana do indukcji reguł akcji dla danych klasyfikacyjnych i opisujących problemy regresyjne.
- W zależności od sposobu prowadzenia indukcji (rozważane podejścia to indukcja z punktu widzenia klasy źródłowej i indukcja z punktu widzenia klasy docelowej) uzyskujemy różne zbiory reguł, o różnych zdolnościach predykcyjnych i opisowych.
- Agregacja wiedzy zawartej w wyznaczonych regułach akcji pozwala na opracowanie algorytmu rekomendacyjnego, zdolnego do wskazania dla zadanego obiektu w jaki sposób należy zmienić wartości jego atrybutów, aby osiągnąć zamierzoną wartość atrybutu decyzyjnego.

## 1.2 Cele pracy

Głównym celem pracy jest opracowanie algorytmu indukcji reguł akcji opierającego się o paradygmat sekwencyjnego pokrywania i uwzględniającego wiodącą rolę przykładów klasy źródłowej lub docelowej w procesie indukcji. Podany algorytm może być stosowany zarówno dla danych klasyfikacyjnych, jak i opisujących problemy regresyjne. Wpływ na postać wynikowych zbiorów reguł akcji będzie mieć także miara jakości zastosowana do nadzorowania indukcji, która to może być swobodnie zadawana przez użytkownika.

Istniejące reguły akcji mogą zostać poddane dalszej eksploracji, co powinno pozwolić na rozwiązanie występujących w ramach zbioru konfliktów i tym samym generację rekomendacji - specjalnych reguł akcji, indukowanych pod zadany przykład. Podanie takiego algorytmu jest dalszym celem pracy.

Wypracowane algorytmy indukcji reguł akcji i rekomendacji, mogą być parametryzowane nie tylko kierunkiem indukcji czy miarą jakości, ale także innymi, zwyczajowymi dla paradygmatu sekwencyjnego pokrywania parametrami. Kolejnym celem pracy jest podanie sposobu postępowania pozwalającego określić zdolności prognostyczne (skuteczność) zbiorów reguł akcji. Zaproponowane postępowanie może zostać wykorzystane do wyboru optymalnych parametrów algorytmu indukcji reguł akcji w danym zadaniu, lub do porównywania między sobą zbiorów reguł akcji uzyskanych przez inne metody odkrywania reguł akcji.

### 1.3 Układ pracy

Niniejsza praca składa się z 6 rozdziałów. Rozdział 1 i 6 to odpowiednio wprowadzenie i podsumowanie pracy. Rozdział 3 jest wprowadzeniem do tematyki indukcji reguł akcji, począwszy od indukcji reguł decyzyjnych wraz z opisem budowy klasyfikatora regułowego i kilku popularnych podejść do indukcji reguł decyzyjnych, w tym metody sekwencyjnego pokrywania. Omówione są również miary jakości reguł klasyfikacyjnych. Następnie zdefiniowane zostają reguły akcji oraz przedstawione jest studium literatury na temat metod ich odkrywania.

Rozdział 3.1 przedstawia pokryciowy algorytm indukcji reguł akcji w zadaniu klasyfikacji w dwóch wariantach: w przód (F) i w tył (B). Indukcja reguł w proponowanym algorytmie nadzorowana jest miarami oceny jakości klasycznych reguł decyzyjnych. W rozdziale 3.3.1 podano modyfikację algorytmu pozwalającą indukować także reguły akcji w zadaniu regresji. Rozdział 3.3.2 krótko omawia reguły akcji w zadaniu analizy przeżycia, bez podawania szczegółów implementacyjnych.

W rozdziale 4 omówiono problemy pojawiające się przy próbach zastosowania reguł akcji, a w szczególności problem wyboru reguły akcji spośród wielu reguł pokrywających pewien przykład. Jako odpowiedź na ten problem podany został algorytm indukcji rekomendacji, będących specjalną formą reguły akcji indukowanej na potrzeby zadanego przykładu (obiektu). Indukcja rekomendacji odbywa się na podstawie struktury danych skupiającej w sobie wiedzę odkrytą przez cały zbiór reguł akcji nazywanej meta-tablicą. Tym samym rodzina algorytmów, nazywana w

pracy modelami akcyjnymi, zostaje rozszerzona o kolejną metodę, dla której dane wejściowe stanowią wyniki działania algorytmów zaproponowanych w rozdziale 3.1. Ostatecznie, w podrozdziale 4.3 wprowadzono sposób weryfikacji reguł akcji, tj. metody pozwalającej ocenić różne modele akcji w kontekście ich przydatności przy przenoszeniu przykładów z klasy niepożądaney do klasy docelowej. Wraz z opisem metody prowadzona jest dyskusja na temat jej istotnych parametrów i interpretacji wyników. Podany sposób oceny jakości modeli akcji jest odpowiedni zarówno dla zadania klasyfikacji jak i regresji.

Opisany w rozdziale 4 sposób oceny zostaje zastosowany w rozdziale 5 do oceny metod opisanych w rozdziałach 3.1 i 4. W podrozdziale 5.1 omówiono parametry algorytmów wykorzystane w badaniach oraz konwencję nazewnictwa badanych wariantów metod. Sekcja 5.2 omawia metody analizy statystycznej stosowane w ocenie algorytmów klasyfikujących, które znajdują zastosowanie w dalszej części rozdziału. Wyniki doświadczenia prezentowane są osobno dla zadania klasyfikacji (podrozdział 5.3) i regresji (5.4). W ramach każdego z tych podrozdziałów najpierw omówiono charakterystykę zbiorów danych testowych, następnie opisano uzyskane wyniki ilościowo i jakościowo, uwzględniając m.in. nie tylko skuteczność modeli akcji ale także ich postać (np. długość i liczbę reguł), by ostatecznie przeprowadzić analizę statystyczną uzyskanych wyników. W rozdziale tym zawarto także parę poglądowych sposobów wizualizacji wyników, które mogą pozwolić łatwiej wybrać najlepszy wariant algorytmu indukcji reguł akcji do danego zastosowania. Celem autora nie jest wykazanie, która metoda jest uniwersalnie najlepsza, ale pokazanie w jaki sposób należy się kierować przy wyborze metody dla własnych celów.

## 2 Podstawowe definicje i metody indukcji reguł klasyfikacyjnych

Reguły uważane są, obok drzew, za najprostszą w zrozumieniu formę reprezentacji wyników algorytmu uczącego się. Stosowane są nie tylko w celach opisu zbioru danych [26], ale także do predykcji [24, 32]. W analizie danych największą popularnością cieszą się reguły klasyfikacyjne [59], zwane także decyzyjnymi, a oprócz nich reguły regresyjne [80], przeżycia [103] oraz akcji [75]. W niniejszym rozdziale przedstawiono podstawowe definicje oraz sposoby reprezentacji reguł decyzyjnych i reguł akcji, a także opisano niektóre algorytmy indukcji reguł.

### 2.1 Sposób reprezentacji reguł

Reguły zapisane słownie mają postać „Jeżeli, to”, jednak zazwyczaj w rozważaniach na ich temat stosuje się zapis o postaci 2.1:

$$\varphi \rightarrow \psi \tag{2.1}$$

W zapisie tym  $\varphi$  nazywana jest przesłanką reguły, zaś  $\psi$  jej konkluzją. Wyrażenie 2.1 należy czytać następująco: „jeżeli spełniona jest przesłanka  $\varphi$ , to spełniona jest konkluzja  $\psi$ ”. Przesłanka ma postać formuły logicznej, natomiast konkluzja  $\psi$  może przybierać różne postaci, w zależności od typu opisywanej reguły. W niniejszej pracy przedstawionych zostanie kilka rodzajów konkluzji: dla reguł decyzyjnych, regresyjnych oraz reguł akcji. Bez względu na postać konkluzji reguły, przesłanka ma postać formuły logicznej, stąd przyjęło się wszystkie reguły postaci 2.1 nazywać mianem reguł logicznych.

**Definicja 1** *Regułą decyzyjną nazywamy formułę postaci:*

$$IF w_1 \wedge w_2 \wedge \dots \wedge w_n THEN C = c_k \tag{2.2}$$

Część reguły decyzyjnej umieszczona po lewej stronie formuły 2.2 nazywa się przesłanką, a część umieszczona po prawej stronie konkluzją reguły. Przesłanka reguły decyzyjnej jest koniunkcją warunków elementarnych  $w_k, k \in \{1, 2, \dots, n\}$ .

**Definicja 2** *Warunek elementarny to wyrażenie postaci:*

$$A_j R V_j, j \in \{1, 2, \dots, |A|\} \quad (2.3)$$

gdzie  $A$  to zbiór atrybutów,  $A_j \in A$ ,  $V_j$  to zakres wartości atrybutu  $A_j$  a  $R$  oznacza operator relacji.

Operator  $R$  we wzorze 2.3 wskazuje na relację zachodzącą między atrybutem  $A_j$  a zakresem wartości  $V_j$  i może przybrać jedną z wartości ze zbioru  $\{\geq, \leq, >, <, \neq, =, \in\}$ . Wobec tego  $V_j$  może przybierać postać podzbioru zbioru wartości atrybutu  $A_j$  lub pojedynczej wartości (np. gdy  $R \equiv \geq$ ). W zastosowaniach praktycznych, ze względu na prostotę reprezentacji, najczęściej przyjmuje się  $R \equiv \in$ . Warunek elementarny należy traktować jako ograniczenie wartości danego atrybutu, tj. aby przesłanka reguły była spełniona, wartości wszystkich atrybutów w niej wymienionych muszą spełniać wszystkie ograniczenia opisane w tej przesłance.

Standardowo – i tak przyjmiemy też w tej pracy – wyróżnia się dwa rodzaje atrybutów. Atrybuty nominalne to atrybuty przyjmujące wartości będące elementami zbiorów dyskretnych i mogą być utożsamiane z cechami jakościowymi opisywanego obiektu. Atrybuty numeryczne przyjmują wartości ciągłe (cechy ilościowe). W przypadku atrybutów nominalnych, wyrażenie 2.3 przyjmuje zazwyczaj postać  $A_j = a_k$ , gdzie  $a_k$  oznacza pewną wartość ze zbioru wartości atrybutu  $A_j$ , rzadziej spotyka się warunki sprawdzające przynależność do podzbioru dziedziny atrybutu nominalnego  $A_j \in \{a_{k_1}, a_{k_2}, \dots\}$ , ze względu na złożoność wyznaczania takich warunków. Dla atrybutów numerycznych często spotykane postacie warunków elementarnych to  $A_j \geq a_k$ ,  $A_j < a_k$  czy  $A_j \in (a_{k_1}, a_{k_2})$ .

Konkluzja reguły 2.2 wskazuje na pewną wartość  $c_k$  z dziedziny atrybutu  $C$ . Atrybut  $C \in A$  nazywamy atrybutem decyzyjnym.

Reguła opisana formułą 2.2 mówi o fakcie zakwalifikowania przykładu do klasy decyzyjnej  $c_k$ , pod warunkiem że spełnione są wszystkie warunki wymienione w jej przesłance.

Reguły decyzyjne znajdują zastosowanie w celu opisanego danych zawartych w pewnym zbiorze przykładów (obserwacji), ale mogą być używane także do klasyfikacji. Dla przykładu wygenerowano reguły decyzyjne na zbiorze danych *car* z repozytorium

UCI [23]. Zbiór ten charakteryzuje 6 atrybutów warunkowych opisujących cechy modeli samochodów oraz atrybut warunkowy określający jak bardzo akceptowalny jest taki samochód przez potencjalnego klienta. Wszystkie atrybuty są nominalne, poniżej opisano ich znaczenie i możliwe wartości:

- **buying** – cena zakupu,  $buying \in \{v\text{-high}, high, med, low\}$
- **maint** – koszty utrzymania,  $maint \in \{v\text{-high}, high, med, low\}$
- **doors** – liczba drzwi,  $doors \in \{2, 3, 4, 5\text{-more}\}$
- **persons** – liczba miejsc dla pasażerów,  $persons \in \{2, 4, more\}$
- **lug\_boot** – pojemność bagażnika,  $lug\_boot \in \{small, med, big\}$
- **safety** – bezpieczeństwo,  $safety \in \{low, med, high\}$

Atrybut decyzyjny przyjmuje jedną z wartości  $\{unacc, acc, good, v\text{-good}\}$ .

Poniżej umieszczono dwie przykładowe reguły decyzyjne możliwe do odkrycia w zbiorze *car*:

$$\begin{aligned} r1 : & \text{IF } doors = 4 \wedge persons = 4 \wedge safety = low \wedge lug\_boot = big \\ & \text{THEN } class = unacc \\ r2 : & \text{IF } doors = 4 \wedge persons = 4 \wedge safety = high \wedge lug\_boot = small \\ & \text{THEN } class = acc \end{aligned} \tag{2.4}$$

Reguły te spełniają funkcję opisową wskazując użytkownikowi związki pomiędzy wartościami poszczególnych atrybutów a wartością atrybutu decyzyjnego, a więc przyporządkowaniem obiektu do reprezentowanego przezeń konceptu. Posiłkując się informacjami o pokryciu poszczególnych reguł można określić jak silne są wskazywane związki. Dodatkowo reguły te mogą zostać użyte do zaklasyfikowania nieznanego przykładu do jednej z klas – metody budowania klasyfikatorów regułowych zostały przedstawione w dalszej części tego rozdziału.

## 2.2 Sposób oceny reguł klasyfikacyjnych. Klasyfikacja za pomocą reguł klasyfikacyjnych

Reguły decyzyjne można uzyskać w wyniku działania wielu algorytmów. Problem odkrywania minimalnych zbiorów reguł opisujących dany zbiór przykładów uznawany

jest za NP-zupełny [3]. Z tego powodu większość algorytmów odkrywania reguł opiera się o heurystyki. Użytkownik systemu regułowego potrzebuje narzędzia, pozwalającego dobrać mu taką metodę odkrywania reguł, która da w efekcie reguły najlepiej nadające się do realizowanego przez niego zadania.

Narzędziem tym są miary oceny jakości reguł (ang. *rule quality measure*) [2,104,108]. Obrazowym przykładem takiej miary jakości może być długość reguły, rozumiana jako liczba warunków elementarnych w jej przesłance. Miara taka byłaby odpowiednia przy porównywaniu ze sobą reguł wygenerowanych przez różne metody w sytuacji, w której użytkownikowi zależy na zwięzłych regułach opisujących pewien zbiór danych. Wówczas im krótsze uzyskane reguły, tym lepiej dla użytkownika, i tym bardziej przydatne wyniki algorytmu. W znakomitej większości przypadków zastosowań systemów regułowych przyjęcie tej miary jako decydującej nie będzie wystarczające – do tej pory podano w literaturze już co najmniej kilkadziesiąt różnych miar, które faworyzują pewne cechy reguł lub ich kombinację, by umożliwić wybór reguł, które najlepiej zaspokoją bieżące potrzeby. Kilka przykładowych miar zostanie opisanych szczegółowo w dalszej części pracy.

W metodach indukcji reguł akcji wprowadzanych w niniejszym opracowaniu szerokie zastosowanie znajdują miary oceny jakości reguł decyzyjnych. W pierwszej części tego podrozdziału opisano podstawowe definicje i przykładowe miary oceny jakości reguł decyzyjnych, druga część przedstawia sposób dokonywania klasyfikacji przy użyciu zbioru reguł decyzyjnych. Trzeba tutaj wspomnieć, że miary oceny jakości reguł znajdują dwojakie zastosowanie. Po pierwsze, używane są do oceniania reguł, by możliwe było porównywanie ich ze sobą, rozwiązywanie konfliktów klasyfikacji, ale także by wśród zbiorów reguł wyszukiwać tych, które są najbardziej interesujące w danym zastosowaniu [2,11]. Drugim obszarem zastosowań jest sterowanie procesem indukcji reguł – wówczas miary stają się częścią algorytmów odkrywania reguł [49,85]. Obydwa te wątki mają duże znaczenie dla metod opisywanych w tej pracy i będą uszczegóławiane w dalszych częściach.

### 2.2.1 Miary oceny jakości reguł klasyfikacyjnych

Miarą oceny jakości reguły jest funkcja  $q$ , która każdej regule w pewnym zbiorze reguł  $R$  przyporządkowuje wartość rzeczywistą (2.5):

$$q : R \rightarrow \mathbb{R} \tag{2.5}$$



	$ [\varphi] $	$ [\neg\varphi] $	$ T $
$ [\psi] $	$p$	$P - p$	$P$
$ [\neg\psi] $	$n$	$N - n$	$N$
$ T $	$p + n$	$P - p + N - n$	$P + N$

**Tabela 2.1:** Tablica kontyngencji

Często dokonuje się także oceny zbioru reguł jako całości. Za przykład takiej miary może służyć średnia długość reguły czy liczba reguł zawarta w zbiorze, ale także dokładność (ang. *accuracy*) czy precyzja (ang. *precision*) całego zbioru reguł, gdy oceniana jest jego zdolność klasyfikacyjna. Dla odróżnienia od miar skupiających się na pojedynczych regułach, funkcję oceny zbioru reguł oznaczamy przez  $Q$ .

Miary oceny jakości dzieli się na obiektywne i subiektywne [40]. W zakresie miar obiektywnych wiele znanych funkcji opartych jest o tablicę kontyngencji reguły, tj. informację o liczbie przykładów pokrywanych przez ocenianą regułę. W tym miejscu należy powiedzieć, że przez „przykład pokrywany przez regułę  $r$ ” rozumiemy taki przykład (obserwację) w pewnym zbiorze danych, który spełnia wszystkie warunki elementarne w przesłance reguły  $r$ . Do miar obiektywnych zaliczamy także miary oparte o analizę postaci reguły, np. długość przesłanki lub rozmiar zakresów warunków elementarnych. Drugą rodzinę miar stanowią miary subiektywne, które to definiowane są tak, aby odzwierciedlić oczekiwania użytkownika względem ocenianych reguł w odniesieniu do problemu, z którym się mierzy. Miary te biorą pod uwagę zarówno analizowane dane, jak i użytkownika danych [35]. Miary te są ściśle związane z analizowanym zbiorem danych i nie da się ich łatwo z powodzeniem zastosować w innym zadaniu, przykładowo w pracy [37] zdefiniowano miarę pozwalającą badać jakość reguł stosowanych do funkcjonalnego opisu genów.

W metodach indukcji reguł wykorzystywane są przede wszystkim miary obiektywne oparte o tablicę kontyngencji. Większość wykorzystywanych miar jest tzw. miarami zysku (ang. *gain measures*). Oznacza to, że im wyższa wartość miary, tym oceniana reguła jest lepsza. Niekiedy spotyka się także miary kosztu (ang. *cost measures*), gdzie relacja między wartością miary a oceną reguły jest odwrotnie proporcjonalna. Miary takie można łatwo sprowadzić do miary zysku poprzez zmianę znaku jej wartości.

W tabeli 2.1 przedstawiono postać wzmiankowanej już w tekście tablicy kontyngencji dla reguły o formie  $r : \varphi \rightarrow \psi$  i pewnego zbioru przykładów  $T$ .

Tablicę kontyngencji buduje się dla pewnej reguły  $r$  i znanego zbioru przykładów  $T$ .  $T$  może być zbiorem treningowym, na którym reguła została wygenerowana. Przedstawiony w 2.1 zapis może wydawać się na pierwszy rzut oka niejasny, jednak w bardzo wygodny sposób pozwala na przedstawienie tablicy kontyngencji zarówno dla problemów binarnych jak i wieloklasowych. Tak przedstawioną tablicę należy czytać jak następuje: symbol  $[\varphi]$  oznacza zbiór przykładów z  $T$  pokrytych przez przesłankę reguły  $r$ .  $[\psi]$  oznacza zbiór przykładów pokrytych przez konkluzję tejże reguły. Symbole  $[\neg\varphi]$  oraz  $[\neg\psi]$  oznaczają zbiory przykładów niepokrywanych przez odpowiednio przesłankę i konkluzję reguły. Mówi się, że przykład jest pokrywany przez przesłankę reguły, gdy spełnia on wszystkie budujące ją warunki elementarne. Analogicznie przykład jest pokrywany przez konkluzję reguły, gdy należy do klasy, na którą wskazuje reguła. Przykłady należące do klasy wskazywanej przez konkluzję reguły nazywane są przykładami pozytywnymi dla tej reguły (należą do klasy pozytywnej), zaś przykłady należące do innej klasy – przykładami negatywnymi (należą do klasy negatywnej). Zastosowanie oznaczenia  $[\neg\psi]$  buduje wygodną abstrakcję i pozwala zaniedbać fakt, że klas negatywnych w zbiorze  $T$  może być wiele.

W komórkach tabeli 2.1 podano liczbę przykładów, jaka znajdzie się w zbiorze, będącym przecięciem zbiorów zapisanych w wierszu i kolumnie tablicy. W tablicy występuje szereg dodatkowych oznaczeń:

- $P$  – liczba wszystkich przykładów pozytywnych w zbiorze  $T$ , tożsame z  $||[\psi]||$
- $N$  – liczba wszystkich przykładów negatywnych w zbiorze  $T$ , tożsame z  $||[\neg\psi]||$
- $p$  – liczba przykładów pozytywnych pokrywanych przez regułę, tj. moc zbioru  $[\psi] \cap [\varphi]$
- $n$  – liczba przykładów negatywnych pokrywanych przez regułę, czyli  $||[\neg\psi] \cap [\varphi]||$

Wartości  $P - p$  oraz  $N - n$  to odpowiednio liczba przykładów pozytywnych i negatywnych niepokrywanych przez regułę. Odnosząc się do tablicy kontyngencji przedstawionej w 2.1, miary obiektywne na niej zdefiniowane można ogólnie przedstawić jako  $q(p, n, P - p, N - n)$ , lub prościej i bez utraty ogólności jako  $q(p, n, P, N)$ . Przy założeniu, że omawia się miarę jakości dla ustalonego zbioru, a więc dla znanych wartości  $P$  i  $N$ , można posunąć się do dalszego skrócenia zapisu do postaci funkcji dwuargumentowej  $q(p, n)$ .

Intuicyjnie, za dobrą regułę uznaje się taką, która pokrywa wiele przykładów pozytywnych (charakteryzuje się wysoką wartością  $p$ ) i jednocześnie jak najmniej

przykładów negatywnych (niska wartość  $n$ ). Miary oceny jakości oparte o tablice kontyngencji na różne sposoby wiążą ze sobą te dwie wartości aby wyrazić tę intuicję w mierzalny sposób, co pozwala na zastosowanie tych miar do sterowania procesem indukcji reguł w algorytmach indukcyjnych. Miary stosowane w algorytmach indukcji reguł powinny dodatkowo charakteryzować się pewnymi własnościami, które w dalszej części zostaną przedstawione dla dwóch popularnych miar: precyzji (ang. *Precision*) i czułości (ang. *Sensitivity*) reguły 2.6.

$$\begin{aligned} Precision(r) &= \frac{p}{p+n} \\ Sensitivity(r) &= \frac{p}{P} \end{aligned} \tag{2.6}$$

Czułość w literaturze angielskiej bywa także oznaczana jako *recall* lub *coverage*. Miara ta wyraża jak dobrze dana reguła pokrywa przykłady pozytywne w stosunku do wszystkich przykładów pozytywnych w danym zbiorze. Precyzja (czasem określana jako *accuracy* lub *confidence*) określa stosunek przykładów pozytywnych do wszystkich przykładów pokrywanych przez regułę. Obydwie te miary przyjmują wartości rzeczywiste w zakresie od 0 do 1 i są miarami zysku. Nie są jednak wolne od wad. Dwie reguły nie pokrywające ani jednego przykładu negatywnego będą przez precyzję ocenione tak samo, niezależnie od tego jak wiele przykładów pozytywnych pokrywają. Czułość natomiast nie uwzględnia w ogóle ile przykładów negatywnych pokrywa reguła. Stąd też bezpośrednie zastosowanie tych miar do sterowania procesem indukcji reguł decyzyjnych nie zawsze pozwalają uzyskać dobre wyniki. Zamiast tego stosuje się miary bardziej rozbudowane, które nierzadko wykorzystują również zależności przedstawione w 2.6, jak np. miara wprowadzona przez Klösgena [51] 2.7

$$Kl\ddot{o}sgen = \left(\frac{p+n}{P+N}\right)^\omega \left(\frac{p}{p+n} - \frac{P}{P+N}\right) \tag{2.7}$$

Łatwo zauważyć, że składnik  $\frac{p}{p+n}$  jest tożsamy z definicją precyzji (2.6). Przy okazji warto nadmienić, że miara 2.7 jest przykładem miary obiektywnej parametrycznej, tj. oprócz danych zawartych w tablicy kontyngencji wykorzystywany jest dodatkowy parametr  $\omega$  zadawany przez użytkownika. Zadowalające wyniki pokryciowej indukcji reguł uzyskuje się przyjmując wartość  $\omega = 0.4323$ , jak wykazali empirycznie Jensen i Fürnkranz [49].

Poniżej przedstawiono definicje kilku miar, które są powszechnie stosowane [104] do sterowania procesem indukcji reguł decyzyjnych. Wymienione miary jakości znajdują

zastosowanie także w dalszej części niniejszej pracy. Miary te zostały wybrane na podstawie empirycznych dowodów wskazujących na ich przydatność w sterowaniu procesem indukcji reguł przedstawionych w literaturze [10, 33, 49, 85, 104].

Miara  $C2 = \frac{Np-Pn}{N(p+n)} \frac{P+p}{2P}$  została wprowadzana w pracy [11] na podstawie doświadczeń jej autora w stosowaniu miary *Coleman*, która stanowi pierwszy człon we wzorze opisującym  $C2$ . Bruha zauważył, że miara *Coleman* kładzie zbyt duży nacisk na dokładność reguł. W mierze  $C2$  drugim członem jest *coverage*, oraz odpowiednie czynniki normalizujące, zapewniające że zakres wartości miary to  $(-\infty, 1)$ , przy czym im bliżej jedności, tym lepiej.

*InfoGain*, skrótowo zwana *Gain*, to miara zapożyczona z algorytmu C4.5, gdzie jest używana jako kryterium podziału przy budowaniu drzewa decyzyjnego [70]. Reprezentuje ona przyrost informacji uzyskiwany dzięki ocenianej regule. Wartość miary można wyznaczyć ze wzoru  $InfoGain = Info(P, N) - Info_{pn}(P, N)$ , przy czym:

$$Info(P, N) = -\left[\frac{P}{P+N} \log_2 \frac{P}{P+N} + \frac{N}{P+N} \log_2 \frac{N}{P+N}\right]$$

$$Info_{pn}(P, N) = \frac{p+n}{P+N} Info(p, n) + \frac{P+N-p-n}{P+N} Info(P-p, N-n).$$

Miara *WeightedLaplace* =  $\frac{(p+1)(P+N)}{(p+n+2)P}$  wprowadza czynnik  $\frac{P+N}{P}$  wiążący jej wartość z rozkładem przykładów pozytywnych i negatywnych do standardowej miary *Laplace*. Jej zadaniem jest estymacja dokładności reguły, z uwzględnieniem korekty dokładności dla reguł, które pokrywają niewiele przykładów. Miara ta skrótowo jest zwana *wLap*, może przyjmować wartości z zakresu  $[0, \infty)$ .

*Correlation* mierzy korelację między rozkładem przykładów pozytywnych i negatywnych pokrywanych przez regułę a ich rzeczywistym przyporządkowaniem do właściwych klas. Zakres zmienności wartości tej miary to  $[-1, 1]$ . Jej wartość można wyznaczyć ze wzoru  $\frac{pN-Pn}{\sqrt{PN(p+n)(P-p+N-n)}}$ .

*RSS* (ang. *Rule Sensitivity and Specificity*) jest różnicą wartości wskaźników znanych jako TPR (ang. *True Positive Ratio*) i FPR (ang. *False Positive ratio*). Jej wartość można wyznaczyć ze wzoru  $RSS = \frac{p}{P} - \frac{n}{N}$ . Zakres wartości tej miary to  $[-1, 1]$ . Im więcej przykładów klasy pozytywnej reguła pokryje, tym lepiej, zaś pokrywanie przez ocenianą regułę przykładów klasy negatywnej wprowadza karę. Zarówno nagroda jak i kara są proporcjonalne do liczby przykładów pozytywnych i negatywnych. Przykładowo, reguła pokrywająca 100% przykładów pozytywnych i ani jednego negatywnego otrzyma ocenę 1. Reguła pokrywająca tylko przykłady negatywne, otrzyma najniższą możliwą ocenę -1, natomiast reguła, która pokryje

100% przykładów pozytywnych i jednocześnie 50% przykładów negatywnych, zostanie oceniona na 0.5.

Miary oceny jakości reguł decyzyjnych stosowane do nadzorowania procesu indukcji reguły powinny promować rozszerzenia przesłanki reguły w sposób, który pozwoli uzyskać zadowalające, potencjalnie bliskie optymalnemu rozwiązanie. W literaturze definiuje się wiele własności, które uznawana za „dobrą” miara oceny jakości powinna spełniać. Ważniejszymi, a jednocześnie bardzo zgodnymi z intuicją własnościami jest monotoniczność miary ze względu na wartości  $p$  i  $n$ .

Pierwsza ze wspomnianych własności mówi, że funkcja  $q(p, n, P-p, N-n)$  powinna być funkcją rosnącą dla rosnącego  $p$  przy niezmiennym się zbiorze przykładów pozytywnych. Formalnie własność tę można przedstawić następująco  $q(p, n, P-p, N-n) < q(p+k, n, P-p-k, N-n), k > 0$ . Wartość miary jakości spełniającej tę własność będzie rosła, gdy rosła będzie pokrycie i dokładność reguły (lub tylko pokrycie, gdy reguła jest dokładna, tj. nie pokrywa żadnego przykładu negatywnego).

Druga własność mówi, że funkcja  $q(p, n, P-p, N-n)$  powinna być funkcją malejącą dla rosnącego  $n$  przy niezmiennym się zbiorze przykładów negatywnych, formalnie:  $q(p, n, P-p, N-n) < q(p, n+k, P-p, N-n-k)$ . Przyjmujemy, że miara jakości powinna maleć w sytuacji w której maleje dokładność przy niezmiennym się pokryciu, tj. pokrywane są przykłady negatywne kosztem braku pokrywania przykładów pozytywnych.

### 2.2.2 Sposób klasyfikacji regułami decyzyjnymi

Zbiór reguł decyzyjnych może zostać wykorzystany do klasyfikacji, tj. do zadania przyporządkowania przykładów (obserwacji) do klasy decyzyjnej. Nie ma przy tym znaczenia, czy analizowany przykład należy do zbioru treningowego. Oczywiście w zastosowaniach praktycznych dokonuje się klasyfikacji przykładów spoza zbioru treningowego, a więc takich, które nie brały udziału w wyznaczeniu zbioru reguł. Łącząc zbiór reguł decyzyjnych ze strategią (schematem) klasyfikacji uzyskuje się klasyfikator regułowy.

Przypisanie przykładu do klasy przez klasyfikator regułowy jest jedynie prognozą na temat wartości atrybutu decyzyjnego. W zastosowaniach praktycznych zazwyczaj nie generuje się reguł na podstawie zbiorów treningowych, które zawierają wszystkie możliwe wartości wszystkich atrybutów. Wobec tego klasyfikacja przykładów spoza zbioru treningowego może być zawodna, gdyż klasyfikator nie posiada pełnej wiedzy

o populacji na temat której wyrokuje odnośnie wartości atrybutu decyzyjnego (klasy). Zawodność procesu klasyfikacji objawi się brakiem pokrycia pewnych przykładów testowych. Między innymi z tego względu niektóre implementacje klasyfikatorów regułowych uwzględniają tzw. regułę domyślną (ang. *default rule*), która posiada pustą przesłankę (pokrywa więc każdy przedstawiony przykład), a w konkluzji wskazuje na klasę, która w zbiorze treningowym występowała najczęściej. Ze względu na dopasowanie reguły domyślnej do każdego możliwego przykładu, musi ona w strategii klasyfikacji być stosowana jako ostatnia.

Jedną ze strategii stosowanych w klasyfikatorach regułowych jest klasyfikacja listą reguł. Utrzymuje się uporządkowaną listę reguł. W trakcie klasyfikacji pewnego przykładu  $t$ , przegląda się kolejno reguły na liście. Konkluzja pierwszej napotkanej reguły, która pokrywa przykład  $t$ , jest odpowiedzią klasyfikatora. Na końcu listy reguł można umieścić regułę domyślną, ale nie jest to obowiązkowe. Wówczas przykład niepokryty przez żadną z reguł na liście nie zostanie sklasyfikowany. Klasyfikator zbudowany według opisanej zasady nazywa się uporządkowanym zbiorem reguł.

**Przykład 2.2.1** *Poniżej zaprezentowano uporządkowaną listę reguł:*

1. *IF*  $a_1 = 3 \wedge a_2 > 15 \wedge a_3 = \text{„OK”}$  *THEN*  $c = \text{„+”}$
2. *IF*  $a_1 = 1 \wedge a_2 < 5 \wedge a_3 = \text{„NOK”}$  *THEN*  $c = \text{„-”}$
3. *IF*  $a_1 = 3 \wedge a_2 > 10 \wedge a_3 = \text{„OK”}$  *THEN*  $c = \text{„+”}$
4. *IF* *THEN*  $c = \text{„+”}$

*W trakcie klasyfikacji przy użyciu tej listy reguł obserwacji o postaci  $((a_1 = 3), (a_2 = 12), (a_3 = \text{„OK”}))$  reguły są przeglądane zgodnie z kolejnością. Pierwsza reguła, która pokrywa przykład, decyduje o odpowiedzi. W tym przypadku zadziała reguła numer 3, wobec czego odpowiedzią klasyfikatora dla tego przykładu będzie przypisanie go do klasy „+”. Na końcu listy reguł umieszczono regułę domyślną, która przypisze niepokryte przez wcześniejsze reguły przykłady również do klasy „+”. Inaczej, klasyfikację z użyciem listy reguł wraz z regułą domyślną można przedstawić jako sekwencję instrukcji warunkowych abstrakcyjnego język programowania:*

Klasyfikacja z zastosowaniem listy reguł ogranicza wiedzę zapisaną w klasyfikatorze regułowym do pierwszej pokrywającej przykład reguły. Może się jednak zdarzyć, że dany przykład  $t$  jest pokrywany przez więcej niż jedną regułę, jednak tylko pierwsza napotkana pokrywająca  $t$  reguła ma pełen wpływ na odpowiedź całego klasyfikatora.

---

```

IF ( $a_1 == 3$  &&  $a_2 > 15$  &&  $a_3 == \text{"OK"}$ ) THEN  $\{c = \text{"+"}\}$ 
ELSE IF ( $a_1 == 1$  &&  $a_2 < 5$  &&  $a_3 == \text{"NOK"}$ ) THEN  $\{c = \text{"-"}\}$ 
ELSE IF ( $a_1 == 3$  &&  $a_2 > 10$  &&  $a_3 == \text{"OK"}$ ) THEN  $\{c = \text{"+"}\}$ 
ELSE  $\{c = \text{"+"}\}$ 

```

---

Wśród algorytmów indukujących listę reguł należy wyróżnić RIPPER [16], modyfikację algorytmu indukcji drzew decyzyjnych C4.5 [70], CN2 [15], Bayesian Rule Lists (BRL) [54] oraz Scalable-BRL [106].

Problem podejmowania decyzji klasyfikującej przez jedną regułę adresują dwa inne popularne schematy klasyfikacji: schemat największego zaufania oraz schemat głosowania, wspólnie określane także jako klasyfikacja nieuporządkowanym zbiorem reguł. Wspólną charakterystyką tych schematów jest konieczność przypisania każdej regule pewnej wartości liczbowej utożsamianej z poziomem zaufania, jakim zostaje ona obdarzona w trakcie klasyfikacji. Wartość tę wyznacza się na zbiorze treningowym, i jest ona odzwierciedleniem jakości reguły – wartością wybranej miary oceny jakości, np. precyzja lub czułość. Poziom zaufania reguły  $r$  będziemy oznaczać jako  $conf(r)$ . W obydwu schematach reguły dodatkowo dzieli się na grupy zawierające reguły wskazujące na tę samą klasę.

W celu sklasyfikowania pewnego przykładu  $t$ , sprawdza się, które reguły go pokrywają. Jeśli przykład  $t$  pokrywany jest przez reguły z jednej tylko grupy, to decyzja klasyfikatora jest taka sama jak reguł zawartych w tej grupie. Jeśli przykład  $t$  nie jest pokrywany przez żadną regułę, można zastosować regułę domyślną lub nie klasyfikować go w ogóle – mówimy wówczas o klasyfikatorze wstrzymującym się [od decyzji] (ang. *abstaining classifier*) [6,68]. Istnieją również rozwiązania, które próbują znaleźć reguły częściowo pokrywające przykład i na tej podstawie podjąć decyzję [38].

Trzecią możliwą sytuacją jest pokrywanie przykładu  $t$  przez reguły z więcej niż jednej grupy, a więc wskazujących na różne klasy decyzyjne, co powoduje konflikt, który rozstrzygany jest na podstawie poziomu zaufania (jakości) reguł.

W schemacie największego zaufania, wśród grup reguł wskazujących na różne klasy wyszukuje się reguły pokrywającej przykład  $t$  o najwyższym poziomie zaufania w danej grupie. Konflikt rozstrzygany jest na korzyść tej klasy, dla której znaleziono pokrywającą przykład  $t$  regułę o poziomie zaufania wyższym, niż dla reguł pokrywających  $t$  w innych grupach. Formalnie, poziom zaufania przynależności klasyfikowanego przykładu  $t$  do klasy decyzyjnej  $c$  wyznacza się zgodnie z 2.8. gdzie  $RUL_c$  oznacza zbiór reguł wskazujących w konkluzji na klasę  $c$ .

$$conf(t, c) = \max\{conf(r) : r \text{ pokrywa } t \wedge r \in RUL_c\} \quad (2.8)$$

W metodzie głosowania konflikt rozstrzygany jest na rzecz tej grupy, która wykaże się najwyższym sumarycznym poziomem zaufania. Poziom zaufania grupy wyznacza się sumując wartości zaufania poszczególnych reguł należących do tej grupy i pokrywających przykład  $t$ . Dodatkowo można zastosować głosowanie proste, w którym każdej regule przyporządkowuje się jednostkowy poziom zaufania (wygrywa wówczas grupa zawierająca najwięcej pokrywających przykład  $t$  reguł) oraz głosowanie ważne, w którym poziom zaufania danej reguły jest wynikiem miary oceny jakości. W obydwu sytuacjach poziom zaufania przynależności przykładu  $t$  do klasy  $c$  wyznacza się zgodnie z 2.9. Jako odpowiedź klasyfikatora dla przykładu  $t$  przyjmuje się tej klasy, dla której uzyskano najwyższy poziom zaufania.

$$conf(t, c) = \sum_{r \in RUL_c \wedge r \text{ pokrywa } t} conf(r) \quad (2.9)$$

Możliwa jest sytuacja, w której nie da się rozwiązać konfliktu, bo np. dwie lub więcej grupy posiadają taki sam poziom zaufania (niezależnie od zastosowanego schematu). Wówczas albo przyporządkowuje się losowo przykład do jednej z klas o najwyższym poziomie zaufania, albo odmawia się klasyfikacji.

Możliwość wyboru funkcji określającej poziom zaufania do reguły pozwala użytkownikowi na dostosowanie klasyfikatora regułowego do specyficznych potrzeb realizowanego zadania. Przykładowo, wybierając jako miarę poziomu zaufania precyzję reguły, w klasyfikacji preferowane będą reguły, który pokrywają mniej przykładów negatywnych.

Wśród algorytmów indukujących nieuporządkowane zbiory reguł, wyróżnić można AQ [59] i jego pochodne, PRISM [12], MODLEM [88], algorytm sekwencyjnego pokrywania stosowany przez Fürnkranza [31] oraz algorytm opisany w podrozdziale 2.3.1.

## 2.3 Wybrane algorytmy indukcji reguł klasyfikacyjnych

W wielu przypadkach celem działania algorytmów odkrywania reguł decyzyjnych jest znalezienie takiego zbioru reguł, który zapewnia poprawne przypisanie przykładów

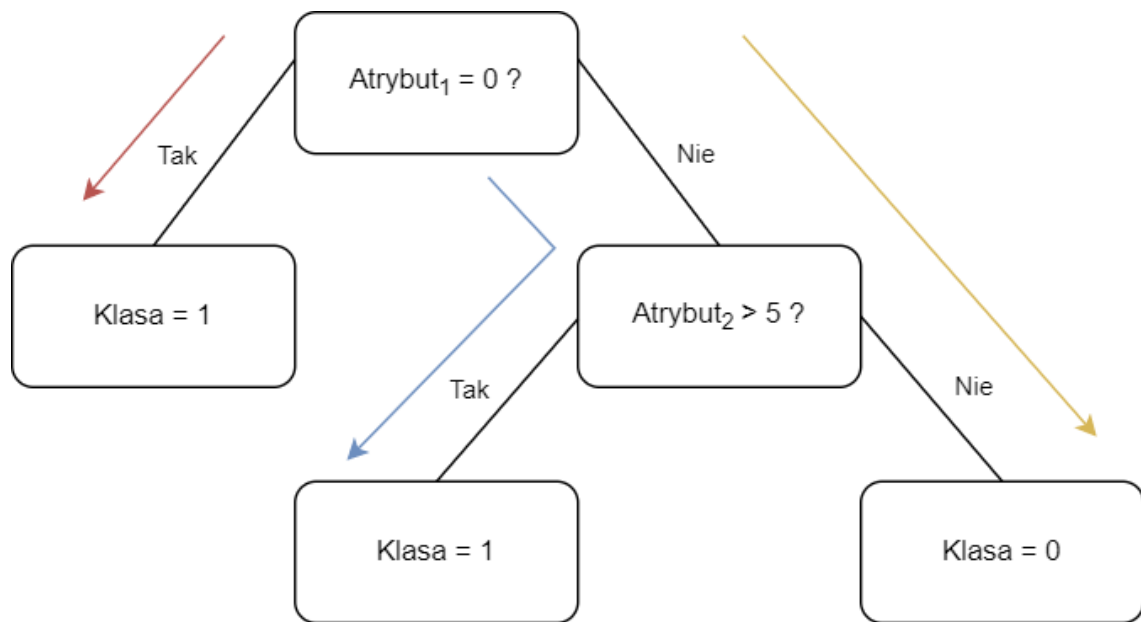


do klas i jednocześnie zawiera tylko tyle reguł, ile potrzeba, tj. najmniejszą możliwą liczbę reguł pozwalającą pokryć wszystkie przykłady. Jest to zadanie kosztowne obliczeniowo [3], dlatego wprowadzono wiele metod heurystycznych, umożliwiających znalezienie dostatecznie dobrych rozwiązań w rozsądnym czasie.

Metody odkrywania reguł akcji, zarówno znane w literaturze jak i przedstawione w tej pracy, są w swych założeniach podobne do algorytmów odkrywania reguł decyzyjnych. Aby umożliwić czytelnikowi zrozumienie tego powiązania, w niniejszej sekcji przedstawiono dwie powszechnie znane strategie generowania reguł decyzyjnych: sekwencyjnego pokrywania oraz analizy zbiorów częstych. Zestaw metod pozwalających odkrywać reguły decyzyjne nie kończy się oczywiście na tych dwóch podejściach, a każdy z paradygmatów manifestowany jest w kodzie wielu algorytmów. Dalsza część rozdziału skupia się na opisanu dwóch najpopularniejszych podejść oraz głębszej analizie dwóch algorytmów. Każdy z podrozdziałów zawiera także wzmianki o innych niż opisane metodach.

Poza opisanymi metodami sekwencyjnego pokrywania i analizy zbiorów częstych, reguły decyzyjne można uzyskiwać na podstawie drzew decyzyjnych. Wystarczy przejść przez każdą ścieżkę z korzenia do liścia takiego drzewa. Wszystkie węzły na ścieżce, wraz ze skojarzonymi z nimi warunkami, zostają zamienione w przesłankę reguły, a decyzja w liściu na końcu ścieżki staje się konkluzją reguły. Proces ten schematycznie przedstawiono na rys. 2.1, gdzie każda kolorowa strzałka obrazuje jedno przejście przez drzewo, dające w efekcie regułę decyzyjną.

Wspomniana metoda konwersji drzew decyzyjnych do postaci reguł znalazła zastosowanie w algorytmie RuleFit [28] stworzonym przez Friedmana i Popescu. RuleFit realizuje ideę baggingu [9] (ang. *bootstrap aggregating*), poprzez indukowanie niezbyt wysokich drzew decyzyjnych na wielu zbiorach utworzonych poprzez losowanie ze zwracaniem nowych na podstawie oryginalnego zbioru treningowego. Drzewa te następnie konwertowane są do postaci reguł decyzyjnych, których decyzja ma wartość „1” gdy analizowany przykład spełnia warunku w przesłance, i „0” w przeciwnym przypadku. W ten sposób reguła może być traktowana jako binarna cecha. Proces ten sprawia, że w kontekście oryginalnego zbioru treningowego są tworzone niejako dodatkowe cechy lub atrybuty, opisujące interakcje zachodzące pomiędzy oryginalnymi atrybutami. Ponieważ drzewa, z których utworzono reguły są „niezbyt wysokie”, również reguły opisują związki między maksymalnie kilkoma atrybutami. Rozwiązanie to przychodzi na myśl tzw. rozszerzenia funkcjonalne Pao w sztucznych sieciach neuronowych [64, 65], jednak użycie reguł pozwala na zbudowanie cech opar-



IF Atrybut<sub>1</sub> = 0 THEN Klasa = 1  
 IF Atrybut<sub>1</sub> != 0 AND Atrybut<sub>2</sub> > 5 THEN Klasa = 1  
 IF Atrybut<sub>1</sub> != 0 AND Atrybut<sub>2</sub> <= 5 THEN Klasa = 0

**Rysunek 2.1:** Uzyskiwanie reguł na podstawie przejść przez drzewo decyzyjne. Reguły uzyskane w kolejnych przejściach przez drzewo oznaczono odpowiadającym kolorem.

tych o bardziej subtelne związki między poszczególnymi atrybutami oryginalnymi. Ostatecznie, na podstawie dodatkowych i oryginalnych cech tworzone są modele liniowe które budują klasyfikator lub model regresyjny.

Obok baggingu niezwykle popularną w ostatnich latach techniką jest boosting [27]. Przykładem algorytmu wykorzystującego to podejście do indukcji reguł decyzyjnych jest ENDER (skrót od ang. *Ensemble of Decision Rules*) [21]. Boosting polega na tworzeniu wielu prostych klasyfikatorów i łączeniu ich w jeden klasyfikator zespołowy (ang. *ensemble classifier*). W przypadku modeli regułowych, rolę prostego klasyfikatora może spełniać pojedyncza reguła. ENDER tworzy reguły iteracyjnie, w sposób uwzględniający zadaną funkcję kosztu, oraz wszystkie do tej pory utworzone reguły. Odpowiedź całego klasyfikatora jest liniową kombinacją decyzji wszystkich reguł. Sposób działania algorytmu ENDER powoduje generację wielu reguł, co

autorzy starają się adresować poprzez ograniczanie z góry liczby reguł lub wybieranie tylko najwyższej ocenionych reguł do stworzenia ostatecznego klasyfikatora.

Znane są także algorytmy odkrywania reguł oparte o metody inspirowane biologicznie, jak algorytmy mrówkowe (ang. *Ant Colony Optimization, ACO*) [56] czy optymalizacja roju cząsteczek (ang. *Particle Swarm Optimization, PSO*) [109]. Żabiński i Zielosko przedstawili metodę odkrywania reguł korzystającą z podejścia zbliżonego do programowania dynamicznego i reprezentacji tablicy decyzyjnej w modelu encja-atrybut-wartość (ang. *Entity-Attribute-Value, EAV*) [111]. Metoda ta pozwala na ograniczenie liczby atrybutów, na podstawie których generowane są reguły. Atrybuty ważone są zgodnie z odchyleniem standardowym ich wartości, gdzie atrybuty o wyższej wartości odchylenia standardowego są preferowane.

Oprócz paradygmatów opisanych bardziej szczegółowo w dalszej części, należy jeszcze wspomnieć o strategiach od szczegółu do ogółu (ang. *bottom-up*) i od ogółu do szczegółu (ang. *top-down*) [32], które najczęściej kojarzone są z metodą sekwencyjnego pokrywania i krokiem specjalizowania reguły, ale dobrze opisują dwa podejścia tworzenia reguł. Pierwsza strategia zakłada tworzenie możliwie najbardziej specyficznych reguł (np. opisujących tylko jeden przykład treningowy) a następnie uogólnianie takich reguł, by były w stanie pokryć więcej przykładów. Drugie podejście oznacza zachowanie odwrotne: najpierw buduje się możliwie ogólne reguły, pokrywające wiele przykładów, a następnie specjalizuje się ich przesłankę, poprawiając ich dokładność, przy jednoczesnym pogorszeniu ogólności. Niezależnie od przyjętej strategii, zawsze zachodzi związek, że im reguła dokładniejsza, tym mniej ogólna, i odwrotnie.

### 2.3.1 Pokryciowy algorytm indukcji reguł decyzyjnych

Metoda sekwencyjnego pokrywania, zwana także strategią pokryciową [31] zalicza się do najpopularniejszych metod indukcji reguł. Ideą tej strategii jest iteracyjne pokrywanie zbioru treningowego – w każdym przebiegu indukowana jest pojedyncza reguła, a przykłady przez nią pokrywane są usuwane ze zbioru podlegającego analizie. Sekwencja ta powtarza się aż do pokrycia wszystkich dostępnych obserwacji. Wynikiem działania metody jest zbiór wszystkich reguł wygenerowanych w tym procesie. Strategię sekwencyjnego pokrywania prezentuje również algorytm 1

Nierzadko warunek stopu głównej pętli algorytmu nie jest tak ostry, jak zaprezentowany w linii 4, lecz przybiera on formę generowania reguł tak długo, jak niepokryty

---

**Algorytm 1** Pokryciowy algorytm indukcji reguł

---

**Wejście:**  $T$  - zbiór przykładów**Wyjście:**  $R$  - zbiór reguł

```

1: function GENERUJREGUŁY( $T$ )
2:    $R \leftarrow \emptyset$ 
3:    $T_{niepokryte} \leftarrow T$ 
4:   while  $T_{niepokryte} \neq \emptyset$  do
5:      $r \leftarrow \text{GENERUJREGUŁĘ}(T, T_{niepokryte})$ 
6:      $T_{niepokryte} \leftarrow T_{niepokryte} \setminus \text{POKRYCIE}(T_{niepokryte}, r)$ 
7:      $R \leftarrow R \cup r$ 
8:   end while
   return  $R$ 
9: end function

```

---

pozostaje pewien procent początkowej liczby przykładów zawartych w zbiorze  $T$  lub tak długo, aż niepokryta pozostanie pewna bezwzględna liczba przykładów.

Wdrożenia zaprezentowanej w Algorytmie 1 strategii sekwencyjnego pokrywania zazwyczaj różnicuje implementacja funkcji *GenerujRegułę*, której wariant prezentuje Algorytm 2. W kontekście indukcji reguł decyzyjnych, funkcja generująca jedną reguła rozbijana jest na dwie fazy. W fazie wzrostu (specjalizacji reguły) przesłanka reguły jest powiększana o kolejne warunki elementarne, w konsekwencji staje się coraz bardziej specyficzna i zaczyna charakteryzować się coraz mniejszym pokryciem. Proces ten dodatkowo jest kontrolowany przez wartość minimalnego pokrycia *mincov* (ang. *minimum coverage*) - wyspecjalizowana reguła musi pokrywać co najmniej *mincov* przykładów pozytywnych, aby zostać zwróconą do dalszego przetwarzania. W następnej fazie reguła jest uogólniana (jej przesłanka jest skracana) w procesie zwanym przycinaniem. Zarówno proces specjalizacji jak i przycinania reguły jest sterowany miarą jakości  $q$ . Indukcja reguł decyzyjnych wymaga nadto wygenerowania reguł dla każdej z klas. Ostatecznie, biorąc pod uwagę przedstawione w tym paragrafie informacje, ogólny schemat sekwencyjnego pokrywania dla reguł decyzyjnych można przedstawić w postaci algorytmu 2. Metodę wykorzystującą procesy specjalizacji i przycinania reguł w literaturze nazywa się podejściem „od góry do dołu” (ang. *top-down*), ze względu na fakt, że początkowo pusta reguła jest coraz to bardziej dopasowywana do przykładów w zbiorze treningowym.

---

**Algorytm 2** Pokryciowy algorytm indukcji reguł decyzyjnych

---

**Wejście:**  $T(A, C)$  - zbiór przykładów opisanych atrybutami  $A$  wraz z wyróżnionym atrybutem decyzyjnym  $C$

**Wejście:**  $q$  - miara oceny jakości

**Wejście:**  $mincov$  - minimalne pokrycie wymagane dla nowej reguły

**Wyjście:**  $R$  - zbiór reguł decyzyjnych

```
1: function GENERUJREGUŁYDECYZYJNE( $T(A, C), q, mincov$ )
2:    $R \leftarrow \emptyset$ 
3:    $\triangleright$  dla każdej klasy w  $C$ 
4:   for  $c \in C$  do
5:      $\triangleright$  zbiór przykładów należących do klasy decyzyjnej  $c$ 
6:      $T_c \leftarrow T[c]$ 
7:     while  $T_c \neq \emptyset$  do
8:        $r \leftarrow$  SPECJALIZUJREGUŁĘ( $T, T_c, q, mincov$ )
9:        $r \leftarrow$  PRZYTNIJREGUŁĘ( $r, T, q$ )
10:       $R \leftarrow R \cup r$ 
11:     end while
12:   end for
13:   return  $R$ 
14: end function
```

---

Specjalizacja reguły polega na dodawaniu kolejnych warunków elementarnych do jej przesłanki, dopóty, dopóki jakość rozszerzanej reguły rośnie. Procedurę tę ilustruje algorytm 3.

Wzrost reguły odbywa się poprzez przeglądanie dostępnych warunków elementarnych, tymczasowe rozszerzanie przesłanki reguły, każdorazową ocenę tak powstałej tymczasowej reguły przez miarę jakości  $q$  oraz wybór najlepszego możliwego – w danym momencie – rozszerzenia reguły. Należy tu zaznaczyć, że ocena tymczasowej reguły dokonywana jest w kontekście całego zbioru treningowego, natomiast test minimalnego pokrycia dotyczy się tylko przykładów pozytywnych dla tej reguły. Jak widać w linii 19 warunkiem stopu wzrostu reguły może być:

- osiągnięcie precyzji reguły równej jednej - mówimy wówczas o regule dokładnej (nie pokrywa ona żadnego przykładu negatywnego)
- brak możliwości znalezienia takiego warunku elementarnego, którego dodanie do przesłanki reguły zwiększyłoby jej jakość i pokryło co najmniej *mincov* dodatkowych przykładów,
- niewygenerowanie ani jednego warunku elementarnego przez funkcję *GenerujWarunkiElementarne*

Tak realizowana strategia optymalizacji przesłanki budowanej reguły pod kątem wartości miary jakości  $q$  jest nazywana metodą wspinaczki (ang. *hill climbing*) i należy ona do klasy algorytmów zachłanych - wybierane jest rozwiązanie (warunek elementarny rozszerzający przesłankę) lokalnie najlepsze.

Dostępne w trakcie wzrostu reguły warunki elementarne są wyznaczone jak następuje (funkcja *GenerujWarunkiElementarne* w algorytmie 3): w podanym zbiorze przykładów przeglądane są wszystkie dostępne atrybuty warunkowe, tj. pomija się atrybut klasowy  $C$ . Jeśli napotkany atrybut jest typu nominalnego, to do ostatecznej listy warunków elementarnych dołączane są warunki postaci *atrybut - wartość* dla wszystkich wartości danego atrybutu, które wystąpiły w analizowanym zbiorze. Jeśli napotkany atrybut jest typu numerycznego, to tworzy się posortowaną listę wartości tego atrybutu, które wystąpiły w przykładach zawartych w analizowanym zbiorze. Następnie wyznacza się średnie arytmetyczne sąsiadujących ze sobą wartości. Do ostatecznej listy warunków elementarnych dopisuje się warunki elementarne postaci *atrybut < wartość* i *atrybut ≥ wartość*, przy czym *wartość* to kolejne z wyznaczonych średnich.

---

**Algorytm 3** Specjalizacja reguły decyzyjnej

---

**Wejście:**  $T(A, C)$  - zbiór przykładów opisanych atrybutami  $A$  wraz z wyróżnionym atrybutem decyzyjnym  $C$

**Wejście:**  $T_c$  - zbiór niepokrytych jeszcze przykładów pozytywnych

**Wejście:**  $q$  - miara oceny jakości

**Wejście:**  $mincov$  - minimalne pokrycie wymagane dla nowej reguły

**Wyjście:**  $r$  - wynikowa reguła

```

1: function SPECJALIZUJREGUŁĘ( $T(A, C), q, mincov$ )
2:   ▷ Pusta reguła wskazująca na klasę  $c$ 
3:    $r \leftarrow (\emptyset \rightarrow C = c)$ 
4:   repeat
5:      $T_r \leftarrow \text{POKRYCIE}(r, T)$ 
6:      $W \leftarrow \text{GENERUJWARUNKIELEMENTARNE}(T_r)$ 
7:      $w_{best} \leftarrow \emptyset$ 
8:      $q_{best} \leftarrow -\infty$ 
9:     for  $w \in W$  do
10:      ▷ Rozszerz przesłankę reguły  $r$  o warunek  $w$ 
11:       $r_t \leftarrow r \wedge w$ 
12:       $T_{r_t} \leftarrow \text{POKRYCIE}(r_t, T_c)$ 
13:      if  $|T_{r_t}| \geq mincov \wedge q(r_t, T) > q_{best}$  then
14:         $w_{best} \leftarrow w$ 
15:         $q_{best} \leftarrow q(r_t, T)$ 
16:      end if
17:    end for
18:     $r \leftarrow r \wedge w_{best}$ 
19:  until  $\text{Precision}(r) = 1 \vee w_{best} = \emptyset$ 
    return  $r$ 
20: end function

```

---

Po fazie wzrostu reguły podlega ona procesowi przycinania, który prezentuje algorytm 4. Przeglądane są kolejne warunki elementarne znajdujące się w przesłance przycinanej reguły  $r$ . Każdy warunek jest tymczasowo usuwany z przesłanki i sprawdzana jest jakość tak zmodyfikowanej reguły na całym zbiorze treningowym  $T$ . Jeśli jakość ta jest równa lub przewyższa oryginalną jakość reguły, warunek ten staje się kandydatem do całkowitego usunięcia z przesłanki, lecz póki co przesłanka nie jest modyfikowana na stałe. Dopiero po przejrzaniu wszystkich dostępnych składników przesłanki, najlepszy kandydat jest usuwany na stałe z przesłanki, a zmodyfikowana w ten sposób reguła ponownie poddawana jest procedurze przycinania. Test w linii 9 sprawdza, czy usunięcie analizowanego warunku spowoduje wzrost lub utrzymanie dotychczasowej wartości miary oceny jakości dla reguły. Wzrost tej wartości po usunięciu warunku jest oczywistą korzyścią. Podobnie jest, jeśli uproszczenie warunku spowoduje utrzymanie oceny przycinanej reguły – oznacza to, że warunek ten nie jest potrzebny i nie ma powodu by komplikował przesłankę reguły.

Przycinanie reguły również stosuje strategię wspinaczki – preferowane przy usunięciu są te warunki, które spowodują maksymalny wzrost jakości reguły, lub ewentualnie brak spadku tej wartości. Przycięte reguły charakteryzują się krótszymi przesłankami i większym pokryciem, co przekłada się na liczbę reguł potrzebnych do pokrycia całego zbioru, a więc rozmiar ostatecznego zbioru reguł.

Ostatecznie przycięta reguła dołączana jest do wynikowego zbioru reguł, pokryte przez nią przykłady usuwane są z bieżącego zbioru, a wzrost i przycinanie nowej reguły jest powtarzane dla niepokrytych jeszcze przykładów.

Sztandarowymi implementacjami metody sekwencyjnego pokrywania do indukcji reguł są AQ [59], CN2 [15], MODLEM [88] oraz popularny (dzięki implementacji w pakiecie WEKA [25]) algorytm RIPPER [17]. Zaprezentowany w algorytmie 2 sposób generacji reguł jest bardziej ogólny niż wymienione implementacje, jako że pozwala na zastosowanie dowolnej miary oceny jakości, umożliwiając w ten sposób znaczny wpływ na sposób generacji reguł i ostateczną postać zbioru reguł.

### 2.3.2 Indukcja reguł decyzyjnych na podstawie strategii przeszukiwania wyczerpującego

Inną rodziną algorytmów indukcji reguł decyzyjnych jest zbiór metod inspirowanych algorytmem Apriori [1], służącym do odkrywania zbiorów częstych i reguł asocjacyjnych. Metody z tej rodziny generalnie realizują strategię przeszukiwania zupełnego



---

**Algorytm 4** Przycinanie reguły decyzyjnej

---

**Wejście:**  $T(A, C)$  - zbiór przykładów opisanych atrybutami  $A$  wraz z wyróżnionym atrybutem decyzyjnym  $C$

**Wejście:**  $r$  - przycinana reguła

**Wejście:**  $q$  - miara oceny jakości

**Wyjście:**  $r'$  - wynikowa, przycięta reguła

```

1: function PRZYTNIJREGULĘ( $r, T(A, C), q$ )
2:    $r' \leftarrow r$ 
3:   repeat
4:      $w_{remove} = \emptyset$ 
5:      $\triangleright$  Bieżąca jakość reguły  $r'$  na zbiorze  $T$ 
6:      $q_{current} = q(r', T)$ 
7:     for  $w \in r'$  do
8:        $q_{remove} = q(r' \setminus w, T)$ 
9:       if  $q_{remove} \geq q_{current}$  then
10:         $w_{remove} \leftarrow w$ 
11:         $q_{current} \leftarrow q_{remove}$ 
12:       end if
13:     end for
14:      $r' \leftarrow r' \setminus w_{remove}$ 
15:   until  $|r'| = 1 \vee w_{remove} = \emptyset$ 
16:   return  $r'$ 
17: end function

```

---

z odrzucaniem reguł, które nie spełniają minimalnych wymagań zdefiniowanych przez użytkownika, podobnie jak Apriori odrzuca w fazie przycinania wszystkie zbiory, które nie są częste. Wśród nich można wyróżnić algorytm *RS-apriori* [53], który bazuje wprost na idei Apriori, oraz algorytm *Explore* [89], który eksploruje dostępną przestrzeń warunków elementarnych celem wykrycia dostatecznie dobrych reguł o coraz to dłuższych przesłankach. Sposób rozbudowy przesłanek jest oparty o ideę rozszerzania zbiorów częstych znaną z metody Apriori. Metoda ta dla zbiorów zawierających tylko atrybuty nominalne została opisana w dalszej części tego podrozdziału. Trzeba tutaj zaznaczyć, że istnieje modyfikacja *Explore*, która umożliwia wyszukiwanie reguł w zbiorach zawierających atrybuty zarówno nominalne, jak i numeryczne [87].

Zasada działania *Explore* została przedstawiona w algorytmie 5. Do zrealizowania całego procesu, potrzebne są dwie procedury pomocnicze, *PrzytnijKandydatów* oraz *Rozszerz*. Procedura *Explore* musi zostać wywołana dla każdej klasy w analizowanym zbiorze przykładów z osobna - pozwala to od razu ustalić konkluzję reguły. Przesłanka jest natomiast budowana na podstawie przekazanego początkowego zbioru warunków elementarnych. Można przyjąć że mają one postać podobną do używanej już w tym rozdziale, tj. (*atrybut = wartość*), w szczególności, że opis dotyczy wersji algorytmu działającego dla zbiorów o atrybutach nominalnych. Aby przygotować początkową listę warunków elementarnych można – po podzieleniu wszystkich przykładów treningowych na grupy pod kątem reprezentowanej przez nie klasy – przeanalizować wszystkie występujące w ramach grupy wartości i utworzyć listę, na której każda z nich występuje tylko raz. Alternatywnie początkowa lista może zostać dostarczona przez eksperta dziedzinowego. Zaletą metody *Explore* jest możliwość zadania warunków stopu, w pseudokodzie oznaczonych jako *SC*. Warunki te pozwalają zatrzymać dalszy rozrost reguł, jeśli np. ich jakość lub postać przestają spełniać kryteria zadane przez użytkownika metody.

W pierwszym kroku, początkowa lista warunków elementarnych jest przycinana przez wywołanie *PrzytnijKandydatów*. Ciało tej procedury przedstawione zostało w algorytmie 6. Trzeba tu nadmienić, że autorzy metody używają terminu *complex*, oznaczającego pojedynczy warunek elementarny lub złożenie dwóch i więcej warunków elementarnych, lub innymi słowy - koniunkcję warunków elementarnych. W ramach działania procedury *PrzytnijKandydatów*, każdy element listy *LS* jest sprawdzany na okoliczność spełniania każdego z warunków stopu zgromadzonych w liście *SC* – jeśli choć jeden z warunków stopu jest spełniony, dana koniunkcja warunków nie może

---

**Algorytm 5** Explore

---

**Wejście:**  $LS$  - zbiór początkowy warunków elementarnych

**Wejście:**  $SC$  - zbiór warunków stopu

**Wyjście:**  $R$  - wynikowy zbiór reguł

```
1: function EXPLORE( $LS, SC$ )
2:    $R \leftarrow \emptyset$ 
3:   PRZYTNIJKANDYDATÓW( $LS, SC, R$ )
4:    $\triangleright$  Kolejka warunków elementarnych
5:    $Q \leftarrow LS$ 
6:   while  $Q \neq \emptyset$  do
7:      $complexes \leftarrow \emptyset$ 
8:      $\triangleright$  Weź pierwszy element z kolejki
9:      $C \leftarrow \text{POP}(Q)$ 
10:    ROZSZERZ( $C, complexes$ )
11:    PRZYTNIJKANDYDATÓW( $complexes, SC, R$ )
12:     $\triangleright$  Umieść nowe complexy na końcu kolejki
13:     $Q \leftarrow Q \cup LS$ 
14:  end while
15:  return  $R$ 
16: end function
```

---

być użyta do rozszerzenia przesłanki i jest usuwana z listy. Następnie mierzona jest precyzja każdego z pozostałych elementów na liście  $LS$ . Jeśli precyzja jest równa 1, oznacza to, że taka koniunkcja warunków elementarnych – wraz z konkluzją ustaloną dla danego przebiegu procedury *Explore* – tworzy regułę dokładną, wobec czego usuwana jest z listy complexów, a umieszczana jest w zbiorze reguł.

---

**Algorytm 6** Przycinanie listy kandydatów

---

**Wejście:**  $LS$  - zbiór początkowy warunków elementarnych

**Wejście:**  $SC$  - zbiór warunków stopu

**Wejście:**  $R$  - dotychczasowe reguły

**Wyjście:**  $R$  - wynikowy zbiór reguł

**Wyjście:**  $LS$  - przycięty zbiór warunków elementarnych

```

1: function PRZYTNIJKANDYDATÓW( $LS, SC, R$ )
2:   for  $c \in LS$  do
3:     if  $c$  spełnia którykolwiek z  $SC$  then
4:        $LS \leftarrow LS \setminus c$ 
5:     else
6:       if  $\text{PRECISION}(c) == 1$  then
7:          $R \leftarrow R \cup c$ 
8:          $LS \leftarrow LS \setminus c$ 
9:       end if
10:    end if
11:  end for
12:  return  $LS, R$ 
13: end function

```

---

Po początkowym przycięciu listy koniunkcji warunków elementarnych, lub inaczej mówiąc przesłanek-kandydatów, tworzona jest kolejka  $Q$  obiektów do dalszego rozszerzania (linia 5 algorytmu 5). Dopóki kolejka ta nie jest pusta, brany jest pierwszy w kolejności element  $c$ , który poddawany jest rozszerzeniu. Rozszerzania wykonywane jest przez procedurę *Rozszerz* formalnie opisaną w algorytmie 7. Uzyskane w ten sposób nowe złożenia warunków poddawane są przycinaniu przez opisaną już metodę *PrzytnijKandydatów*, a wynikowa lista dodawana jest na koniec kolejki  $Q$ . Wyczerpanie elementów w kolejce oznacza zakończenie głównej pętli algorytmu i zwrócenie wszystkich odkrytych reguł.

Więcej uwagi wymaga opis procedury *Rozszerz* (Algorytm 7). Przetwarzana w niej przesłanka-kandydat jest rozszerzana, co pozwala na stworzenie nowego zbioru kandydujących przesłanek. Rozszerzania przesłanki polega na dołączaniu do obecnie przetwarzanej koniunkcji warunków elementarnych  $C$  nowych warunków elementarnych, które w nim jeszcze nie wystąpiły (por. uwaga o indeksie  $h$  w pseudokodzie). Przykładowo, załóżmy że początkowa lista warunków elementarnych to  $\{c_1, c_2, c_3\}$  i że żaden z nich nie został odsiany w początkowym wywołaniu *PrzytnijKandydatów*. Procedura *Rozszerz* wywoływana jest dla warunku  $c_1$ . W wyniku uzyskamy następujący zbiór przesłanek:  $\{c_1 \wedge c_2, c_1 \wedge c_3\}$ .

---

**Algorytm 7** Rozszerzanie przesłanek

---

**Wejście:**  $C$  - rozszerzana przesłanka

**Wyjście:** *complexes* - uzyskane rozszerzenia przesłanki  $C$

```

1: function ROZSZERZ( $C$ )
2:    $k \leftarrow$  długość przesłanki  $C$ 
3:    $h \leftarrow$  najwyższy indeks przesłanek składowych  $C$ 
4:   complexes  $\leftarrow$   $\{C \wedge s_{h+1}, C \wedge s_{h+2}, \dots, C \wedge s_n\}$ ,
      gdzie  $h$  jest najwyższym indeksem warunków elementarnych występujących w  $C$ 
   return complexes
5: end function

```

---

Po zakończeniu procedury *Rozszerz* uzyskane koniunkcje warunków elementarnych poddawane są filtrowaniu w procedurze *PrzytnijKandydatów*. Obiekty które pozostaną na liście po wykonaniu tej procedury, są dołączane na koniec głównej kolejki  $Q$ . Sposób rozszerzania przesłanek kandydujących przypomina metodę budowania coraz to dłuższych zbiorów częstych w metodzie Apriori.

W najgorszym przypadku metoda *Explore* działa w czasie eksponencjalnym ze względu na liczbę warunków elementarnych. Jak zaznaczają jej autorzy, zazwyczaj użytkownicy zainteresowani są krótkimi i silnymi regułami, a te - przy założeniu, że kolejka  $Q$  jest uporządkowana względem „siły” poszczególnych atrybutów - zostaną wygenerowane jako pierwsze w kolejności, co może pozwolić na wcześniejsze przerywanie obliczeń. W przeciwieństwie do metody sekwencyjnego pokrywania, algorytm *Explore* generuje **wszystkie** reguły spełniające wymagania użytkownika, a nie *tylko* dostatecznie dużo reguł wystarczających do pokrycia całego zbioru treningowego.

## 2.4 Reguły akcji

W sekcji 2.1 przedstawiono opis reguł decyzyjnych oraz obszary ich zastosowania, takie jak odkrywanie wiedzy, opisywanie zbioru danych czy zadanie klasyfikacji - są to obszary analizy danych, w których reguły decyzyjne są chętnie wykorzystywane, szczególnie w zadaniach, w których wymagana jest klarowność uzyskanego opisu danych. Reguły, stosowane jako narzędzie opisu zbioru danych, są stosowane także w zadaniach regresji i analizy przeżycia, w których mogą być także używane jako narzędzie predykcyjne [82]. Z jednej strony możemy traktować zbiór reguł jako skompresowany opis wiedzy zawartej w analizowanym zbiorze danych, z drugiej zaś jako środek pozwalający nam przewidywać wartość atrybutu decyzyjnego dla przykładów nieznanymi w momencie indukcji reguł.

Innym interesującym aspektem analizy danych jest ustalenie, jakie warunki muszą zostać spełnione, aby pewien przykład mógł być zaklasyfikowany do klasy innej niż jego obecne przyporządkowanie. Przykładowo, projektant samochodów mógłby być zainteresowany analizą wspomnianego już zbioru *car* pod kątem znalezienia takiej cechy modeli pojazdów które nie są chętnie kupowane (należą do klasy *unacc*), której ulepszenie w nowym wydaniu sprawi, że odświeżony model będzie bardziej pożądanym przez kupujących (przejdzie do klasy *acc*, *good* lub *v-good*). Rodzi to potrzebę zapisania wymaganych zmian (zwanymi w dalszej części pracy rekomendacjami), mówiących jakie wartości przykładów należy zmienić, aby możliwe było zaliczenie ich do innej niż pierwotna klasa. Formuły logiczne opisujące takie przejścia nazywa się **regułami akcji** [75].

**Definicja 3** Rozważmy dwie reguły decyzyjne  $r_1$  i  $r_2$  o postaci 2.10

$$\begin{aligned} r_1 : IF w_{s_1} \wedge w_{s_2} \wedge \dots \wedge w_{s_n} THEN C = c_s \\ r_2 : IF w_{t_1} \wedge w_{t_2} \wedge \dots \wedge w_{t_n} THEN C = c_t \end{aligned} \quad (2.10)$$

przy czym  $w_{s_i} \equiv A_i \ R \ V_{s_i}$ ,  $w_{t_i} \equiv A_i \ R \ V_{t_i}$ . Równoważny zapis tych reguł zgodnie z 2.3:

$$\begin{aligned} r_1 : IF (A_1 \in V_{s_1}) \wedge (A_2 \in V_{s_2}) \wedge \dots \wedge (A_n \in V_{s_n}) THEN C = c_s \\ r_2 : IF (A_1 \in V_{t_1}) \wedge (A_2 \in V_{t_2}) \wedge \dots \wedge (A_n \in V_{t_n}) THEN C = c_t \end{aligned} \quad (2.11)$$

Przyjmijmy, że klasy wskazywane przez te reguły są różne, tj.  $s \neq t$ . Złączenie reguł  $r_1$  i  $r_2$  przedstawione na 2.12 nazywamy regułą akcji:

$$IF (A_1, V_{s_1} \rightarrow V_{t_1}) \wedge (A_2, V_{s_2} \rightarrow V_{t_2}) \wedge \dots \wedge (A_n, V_{s_n} \rightarrow V_{t_n}) THEN C = c_s \rightarrow c_t \quad (2.12)$$

Formułę 2.12 można bez utraty ogólności zapisać w uproszczonej formie przedstawionej poniżej:

$$\text{IF } w_{s_1} \rightarrow w_{t_1} \wedge w_{s_2} \rightarrow w_{t_2} \wedge \dots \wedge w_{s_n} \rightarrow w_{t_n} \text{ THEN } C = c_s \rightarrow c_t \quad (2.13)$$

Reguła akcji, podobnie jak reguła decyzyjna, dzieli się na przesłankę i konkluzję. Konkluzja reguły akcji zawiera pojedynczą **akcję elementarną** dotyczącą atrybutu decyzyjnego, natomiast przesłanka składa się z koniunkcji akcji elementarnych.

**Definicja 4** *Akcją elementarną nazywamy formułę następującej postaci 2.14*

$$(A_k, V_{s_k}) \rightarrow (A_k, V_{t_k}) \quad (2.14)$$

którą można w skróconej formie zapisać jak w 2.15

$$(A_k, V_{s_k} \rightarrow V_{t_k}) \quad (2.15)$$

gdzie  $A_k$  to pewien atrybut warunkowy, zaś  $V_{s_k}$  i  $V_{t_k}$  są zakresami tego atrybutu.

- $V_{s_k}$  nazywa się zakresem źródłowym akcji elementarnej,
- $V_{t_k}$  nazywa się zakresem docelowym akcji elementarnej,
- $A_k \rightarrow V_{s_k}$  nazywa się przesłanką akcji elementarnej,
- $A_k \rightarrow V_{t_k}$  nazywa się konkluzją akcji elementarnej.

Akcję elementarną można w sposób uproszczony zapisać jako 2.16

$$w_{s_k} \rightarrow w_{t_k} \quad (2.16)$$

gdzie  $w_{s_k}$ ,  $w_{t_k}$  są odpowiednio źródłowym i docelowym warunkiem elementarnym.

Akcję elementarną należy rozumieć jako wskazówkę dotyczącą zmiany wartości atrybutu  $A_k$  z wartości źródłowej  $V_{s_k}$  do wartości docelowej  $V_{t_k}$ . Należy przy tym zwrócić uwagę, że akcja elementarna może być zdefiniowana zarówno dla atrybutu numerycznego jak i nominalnego. W pierwszym przypadku wartości źródłowa i docelowa mogą być dowolnymi rodzajami zakresów, dla wygody jednak warto przyjąć, że są to zawsze przedziały wartości jakie powinien spełnić dany atrybut. W przypadku atrybutów nominalnych zakres ten można być dowolnym podzbiorem zbioru wartości atrybutu, zazwyczaj jest to zbiór jednoelementowy. Z definicji akcji elementarnej wprost wynika fakt, że oba zakresy znajdujące się w akcji elementarnej muszą być zdefiniowane na podstawie tego samego atrybutu warunkowego.

**Definicja 5** Załóżmy, że dana jest akcja elementarna  $a : (A_k, V_{s_k} \rightarrow V_{t_k})$  oraz pewien zbiór przykładów  $T$ .

Wykonanie akcji elementarnej to realizacja takich działań, które spowodują, że dla każdego takiego przykładu  $t \in T : A_k(t) \in V_{s_k}$  nastąpi taka zmiana wartości, że  $A_k(t) \in V_{t_k}$ .

Przesłanka reguły akcji jest koniunkcją akcji elementarnych, przez wykonanie przesłanki reguły akcji należy więc rozumieć wykonanie - zgodnie z podaną wyżej definicją - wszystkich akcji elementarnych budujących tę przesłankę.

**Definicja 6** Dana jest reguła akcji  $r$ . Wprowadza się następujące oznaczenia:

- $r_s$  - to reguła decyzyjna, której przesłanka zbudowana jest z przesłanek akcji elementarnych reguły  $r$ , a jej konkluzja wskazuje na klasę źródłową. Regułę taką nazywa się regułą źródłową reguły akcji  $r$ .
- $r_t$  - to reguła decyzyjna, której przesłanka zbudowana jest z konkluzji akcji elementarnych reguły  $r$ , a jej konkluzja wskazuje na klasę docelową. Regułę taką nazywa się regułą docelową.
- $W_r$  - zbiór wszystkich akcji elementarnych w przesłance  $r$
- $W_{r_s}$  - zbiór wszystkich warunków elementarnych w przesłance  $r_s$
- $W_{r_t}$  - zbiór wszystkich warunków elementarnych w przesłance  $r_t$

Dla zbioru reguł akcji  $R$  definiuje się następujące zbiory:

- $W_R = \bigcup_{r \in R} W_r$ ,
- $W_{R_s} = \bigcup_{r \in R} W_{r_s}$ ,
- $W_{R_t} = \bigcup_{r \in R} W_{r_t}$ .

Konkluzję reguły akcji stanowi akcja elementarna postaci  $C = c_s \rightarrow c_t$ , odnosząca się do atrybutu decyzyjnego, a więc do atrybutu typu nominalnego. Wyraża ona zmianę klasyfikacji obiektu z przypisania do klasy źródłowej  $c_s$  na przypisanie do klasy docelowej  $c_t$ , pod warunkiem że wykonane zostaną wszystkie akcje elementarne zdefiniowane w przesłance reguły. Konkluzję reguły akcji można zapisać także zgodnie z definicją jako  $(C, \{c_s\} \rightarrow \{c_t\})$ , jednak zapis użyty w formułach 2.12 i 2.13 jest czytelniejszy, a nie powoduje utraty ogólności.



Zgodnie z definicją 3, regułę akcji można traktować jako złożenie dwóch reguł klasyfikacyjnych. Dla każdej z takich reguł składowych, nazywanych źródłową i docelową, można zmierzyć pokrycie na pewnym zbiorze przykładów. Dla rozróżnienia od oznaczeń wprowadzonych w podrozdziale 2.2.1 wprowadza się następujące oznaczenia:

- $p_s$  - liczba przykładów klasy źródłowej pokrywanych przez regułę źródłową,
- $n_s$  - liczba przykładów klas innych niż źródłowa pokrywanych przez regułę źródłową,
- $p_t$  - liczba przykładów klasy docelowej pokrywanych przez regułę docelową,
- $n_t$  - liczba przykładów klas innych niż docelowe pokrywanych przez regułę docelową,
- $P_s$  - liczba przykładów klasy źródłowej,
- $N_s$  - liczba przykładów klas innych niż źródłowa,
- $P_t$  - liczba przykładów klasy docelowej,
- $N_t$  - liczba przykładów klas innych niż docelowa,

Oczywiście w przypadku problemu dwuklasowego zachodzi  $P_s = N_t$  i  $N_s = P_t$ .

**Przykład 2.4.1** Załóżmy prostą tablicę decyzyjną jak poniżej:

$a$	$b$	$d$
$a_1$	$b_1$	$d_1$
$a_1$	$b_2$	$d_1$
$a_2$	$b_2$	$d_2$

Przyjmijmy, że atrybut  $d$  jest atrybutem decyzyjnym.

Reguła akcji o postaci:

$IF (a, a_1 \rightarrow a_2) AND (b, b_1 \rightarrow b_2) THEN (d, d_1 \rightarrow d_2)$

w tej tablicy decyzyjnej będzie charakteryzować się następującymi wartościami:  $p_s = 1, n_s = 0, p_t = 1, n_t = 0, P_s = N_t = 2, N_s = P_t = 1$ .

Konkluzja reguły akcji nie jest jedynym szczególnym przypadkiem postaci akcji elementarnej. Pozostałe możliwości (wraz z nazwami umożliwiającymi łatwą identyfikację przypadków szczególnych) opisano poniżej, przy założeniu, że akcja oparta jest o atrybut numeryczny.

1. Akcja elementarna postaci  $(A_k, (a_1, b_1] \rightarrow (a_2, b_2])$ , przy czym  $a_1 \neq a_2, b_1 \neq b_2$ , na przykład  $(temperatura, (36, 37] \rightarrow (40, 42])$ .
2. Akcja pusta:  $(A_k, (a_1, b_1] \rightarrow nil)$ . Wykonanie takiej akcji nakazuje dla danego przykładu ograniczenie zakresu źródłowego, ale nie narzuca zakresu docelowego, np.  $(temperatura, (36, 37] \rightarrow nil)$ . Sens istnienia akcji elementarnych tej postaci uwypukla się przy dłuższych przesłankach reguł, gdy są one używane do wyselekcjonowania przykładów źródłowych, dla których pozostałe akcje elementarne będą wskazywać wymagane zmiany wartości atrybutów.
3. Akcja zawężająca  $(A_k, \emptyset \rightarrow (a_2, b_2])$ . Wykonanie takiej akcji nakazuje sprawdzenie wartości atrybutu  $A_k$  do wartości sprecyzowanej zakresem docelowym, niezależnie od wartości źródłowej atrybutu, na przykład  $(temperatura, \rightarrow (36, 37])$ .
4. Akcja podtrzymująca  $(A_k, (a_1, b_1] \rightarrow (a_1, b_1])$ . Wykonanie takiej akcji polega na utrzymaniu źródłowego zakresu po zakończeniu działań wpływających na przykład. Akcję tego typu można zapisać w uproszczeniu jako  $(A_k, (a_1, b_1])$ , a więc w postaci tożsamej warunkowi elementarnemu. W porównaniu do punktu drugiego, akcja ta prezentuje inne znacznie. Akcja pusta nie ogranicza w ogóle wartości na atrybucie dla przykładów klasy docelowej, a akcja podtrzymująca wymaga, by znajdowały się w ustalonym zakresie. Innymi słowy, akcja zawężająca po stronie źródłowej pokrywa dowolny przykład, a akcja podtrzymująca tylko taki, który jest pokrywany przez zadany zakres wartości źródłowej.

Akcje puste, zawężające i podtrzymujące zyskują na znaczeniu jako część dłuższej przesłanki reguły akcji, i mają wpływ na sposób oceny reguł składowych.

**Przykład 2.4.2** Zakładając taką samą tablicę decyzyjną jak w przykładzie 2.4.1.

*Reguła IF  $(a, a_1 \rightarrow a_2)$  AND  $(b, b_1 \rightarrow b_2)$  THEN  $(d, d_1 \rightarrow d_2)$*

*zawiera w przesłance pełną akcję elementarną i można ją scharakteryzować następującymi wartościami:  $p_s = 1, n_s = 0, p_t = 1, n_t = 0$ .*

*Reguła IF  $(a, a_1 \rightarrow nil)$  AND  $(b, b_1 \rightarrow b_2)$  THEN  $(d, d_1 \rightarrow d_2)$*

*zawiera w przesłance akcję pustą, co wpływa następująco na jej charakterystykę:  $p_s = 1, n_s = 0, p_t = 1, n_t = 1$ .*

*Reguła IF  $(a, a_1 \rightarrow a_2)$  AND  $(b, nil \rightarrow b_2)$  THEN  $(d, d_1 \rightarrow d_2)$  ma w przesłance akcję zawężającą, w związku z czym  $p_s = 2, n_s = 0, p_t = 1, n_t = 0$ .*

## 2 Podstawowe definicje i metody indukcji reguł klasyfikacyjnych

Przesłanka reguły IF  $(a, a_1 \rightarrow a_1)$  AND  $(b, b_1 \rightarrow b_2)$  THEN  $(d, d_1 \rightarrow d_2)$  zawiera akcję podtrzymującą. Dla tego przypadku mamy  $p_s = 1, n_s = 0, p_t = 0, n_t = 1$ .

W kontekście reguł akcji atrybuty dyskryminuje się nie tylko ze względu na ich postać numeryczną lub nominalną, lecz także ze względu na zdolność do zmieniania wartości danego atrybutu. Wyróżnia się atrybuty **stałe** (ang. *stable*) oraz **zmienne** (ang. *flexible*). Wartość atrybutu stałego nie może ulec zmianie w wyniku żadnych operacji, natomiast wartość atrybutu zmiennego może być dowolnie zmieniana. Przykładowo, atrybutem stałym będzie „rok urodzenia” lub „płeć”, a atrybutem zmiennym „oprocentowanie pożyczki” lub „przychód”. Zmienność i stałość atrybutów jest specyficzna dla każdego rozpatrywanego zbioru danych, np. w pewnych zadaniach dopuszczalne będzie sugerowanie zmiany wartości atrybutu „przychód”, a w innych nie.

Konsekwencją rozróżnienia atrybutów na stałe i zmienne jest fakt, że w oparciu o atrybut stały nie można definiować dowolnych akcji elementarnych, a jedynie akcje podtrzymujące czyli warunki elementarne, lub ewentualnie akcje puste. Oznacza to przykładowo, że w regułach pokazanych w 2.12 pewne dwa warunki elementarne są sobie równoważne, tj. dla pewnego  $i$ ,  $w_{s_i} \equiv w_{t_i}$ . Sensem definiowania takich akcji jest ograniczenie przeszukiwanej przestrzeni atrybutów wśród przykładów należących do klas  $c_s$  i  $c_t$ . Ogólną postać reguły zawierającej w przesłance akcje elementarnego tego typu przedstawiono w 2.17.

Oczywiście, akcje oparte o atrybuty zmienne również ograniczają przestrzeń przeszukiwanego atrybutu wśród reprezentantów co najmniej jednej z klas zawartych w zbiorze, jednak w przypadku atrybutów zmiennych dopuszczalne (a wręcz pożądane) jest, by budowane akcje były pełne, dzięki czemu reguła może w swojej przesłance prezentować wiedzę o niezbędnych zmianach wartości tych atrybutów, co jest jednym z celów indukcji reguł akcji. Innymi słowy, akcje (warunki elementarne) oparte o atrybuty stałe, ograniczają przykłady, co do których mają zastosowanie pozostałe akcje oparte o atrybuty zmienne znajdujące się w przesłance reguły.

$$\text{IF } w_{s_1} \wedge \dots \wedge w_{s_i} \rightarrow w_{t_i} \wedge \dots \text{ THEN } C = c_s \rightarrow c_t \quad (2.17)$$

Wyrażenie 2.17 należy rozumieć następująco: obserwacja może zmienić klasę z  $c_s$  na  $c_t$ , pod warunkiem że spełnia wszystkie warunki elementarne oraz zostaną wykonane wszystkie akcje wymienione w przesłance reguły.

Przykładową regułą akcji, będącą złożeniem reguł pokazanych w 2.4 zaprezentowano poniżej:

$$\begin{aligned} \text{IF } doors = 4 \wedge persons = 4 \wedge safety = low \rightarrow high \wedge lug\_boot = big \rightarrow small \\ \text{THEN } class = unacc \rightarrow acc \end{aligned} \quad (2.18)$$

Reguła akcji w 2.18 zawiera dwie akcje elementarne, które upraszczają się do warunku elementarnego oraz dwie akcje elementarne wyróżniające zakresy po lewej i prawej stronie. Cała reguła została zapisana w sposób uproszczony, łatwiejszy w odbiorze dla czytelnika – pominięte zostały nawiasy oraz dodany został znak równości. Zapis zgodny z definicją 4 przedstawiono na 2.19

$$\begin{aligned} \text{IF } (doors, 4 \rightarrow 4) \wedge (persons, 4 \rightarrow 4) \wedge (safety, low \rightarrow high) \\ \wedge (lug\_boot, big \rightarrow small) \text{ THEN } (class, unacc \rightarrow acc) \end{aligned} \quad (2.19)$$

## 2.5 Wybrane algorytmy indukcji reguł akcji

Reguły akcji i metody ich indukcji pojawiają się w literaturze od roku 2000 [75]. Dzięki zwartej i formalnej postaci reguł akcji, pozwalającej opisywać możliwe przejścia obiektów między klasami jako formuła logiczna, stały się one narzędziem odkrywania i opisu wiedzy o takich zależnościach. Z drugiej strony znajdują zastosowanie, gdy pojawia się potrzeba podjęcia fizycznych działań celem osiągnięcia zamierzonego efektu. Automatyczne odkrywanie takich rekomendacji, niekoniecznie z zastosowaniem metod regułowych, bywa w literaturze nazywane *action mining* [34, 43, 78, 97]. Niniejsza praca skupia się na obydwu przedstawionych zagadnieniach, przez co przeważającą część jej objętości poświęcono opisom reguł, reguł akcji i metod ich indukcji, gdyż zgodnie z obecną wiedzą, tylko metody regułowe są w stanie jednocześnie być pomocne w osiągnięciu obu wspomnianych celów. W kontekście reguł akcji można także omówić starania odkrywania „użytecznej wiedzy” (ang. *actionable knowledge*), których wyniki można przestawić także w postaci reguł akcji. Mówi się, że wzorzec zawiera użyteczną wiedzę, gdy użytkownik może wykorzystać ją do podjęcia działania, które przyniesie mu korzyść w pewnej dziedzinie [67].

Bliski związek reguł akcji z regułami decyzyjnymi powoduje, że nierzadko algorytmy indukcji reguł akcji bazują na znanych wcześniej sposobach odkrywania reguł

klasyfikacyjnych. Naturalnym podejściem wydaje się być inspirowanie sprawdzonymi strategiami do rozwiązania podobnych problemów. Trzy główne podejścia do indukcji reguł akcji to odkrywanie ich wprost w tablicy decyzyjnej, eksploracja przestrzeni przykładów metodami wyszukiwania wyczerpującego oraz tworzenie reguł akcji na podstawie wcześniej znanych reguł klasyfikacyjnych lub drzew decyzyjnych.

W kolejnych sekcjach krótko przedstawiono działanie metod reprezentujących każde ze wspomnianych podejść, szerszy opis wielu metod odkrywania reguł akcji można znaleźć w przeglądowej pracy autorstwa Dardzińskiej [19].

### 2.5.1 Indukcja reguł akcji wprost z tablicy decyzyjnej

Pojęcie reguł akcji zostało wprowadzone przez Rasia równocześnie z metodą pozwalającą na ich odkrywanie [75]. W metodzie tej wykorzystano pojęcie systemu informacyjnego [66], teorię zbiorów przybliżonych i reprezentację danych w postaci tablicy decyzyjnej, z której na podstawie odkrytych reduktów decyzyjnych (d-reduktów) tworzone są reguły decyzyjne, przy czym do dalszego przetwarzania wybierany jest d-redukt zbudowany w oparciu o relatywnie (w odniesieniu do całego zbioru atrybutów) mało atrybutów stałych. Następnie, wszystkie możliwe pary wybranych d-reduktów są przeszukiwane pod kątem znalezienia par, które są zbudowane w oparciu o podobne zbiory atrybutów. Oba testy przeprowadzane na d-reduktach mogą być sterowane przez użytkownika poprzez modyfikację wartości wag przekazywanych do algorytmu. Na podstawie wyselekcjonowanych par d-reduktów budowane są reguły decyzyjne. W celu uzyskania z nich reguł akcji, badane są wszystkie pary reguł o sprzecznych decyzjach. Jeśli badana para jest nieodróżnialna ze względu na zbiór atrybutów stałych zawartych w przesłankach obu reguł można z niej utworzyć regułę akcji. W tym rozwiązaniu reguła akcji będzie zawierać tylko atrybuty zmienne mające różne wartości w przesłankach reguł tworzących parę. Należy też zaznaczyć, że metodę zaprojektowano do analizy danych klientów banku, w którym klasy abstrakcji odzwierciedlały zysk banku, jakim charakteryzuje się dana grupa klientów. Z tego względu dodatkowym kryterium wyboru reguł decyzyjnych do budowy reguły akcji jest to, aby klasy w konkluzjach reguły akcji zachowywały porządek - inaczej mówiąc dozwolone jest tylko generowanie reguł akcji, które wskazują na przejście klienta z grupy dającej mniejszy zysk do takiej, która pozwoli zysk powiększyć. Metoda ta pośrednio wykorzystuje reguły decyzyjne, jednak nie są one zadawane jako dane

wejściowe algorytmu przez użytkownika, co pozwala uznać ją za metodę odkrywającą reguły akcji wprost z tablicy decyzyjnej.

Pierwsza metoda odkrywania reguł akcji zawiera wiele obostrzeń i założeń co do analizowanych danych, jak np. konieczność istnienia relacji porządku między wartościami atrybutu decyzyjnego czy wymóg, by wszystkie atrybuty były dyskretne. Problemy te zostały rozwiązane w innej metodzie autorstwa Rasia i Im: ARED [46] (ang. *Action Rule Extraction from a Decision Table*), która jest zdolna do indukcji reguł akcji wprost z tablicy decyzyjnej, w której wyróżniono atrybuty stałe, zmienne i jeden atrybut decyzyjny  $d$ .

Do działania algorytmu niezbędne są dwa zbiory,  $\tau$  i  $\delta$ , służące do badania możliwych przejść w analizowanej tablicy decyzyjnej. Zbiory te definiowane są następująco:

$$\begin{aligned} \tau &= T \cdot d_1, \text{ gdzie } d_1 \in V_d(\forall \rho_1 \in T \cdot d_1)(sup(\rho_1) \geq \lambda_1) \\ \delta &= T \cdot d_2, \text{ gdzie } d_2 \in V_d(\forall \rho_2 \in T \cdot d_2)(sup(\rho_2) \geq \lambda_1) \end{aligned} \quad (2.20)$$

Przez  $T$  należy rozumieć zbiór wszystkich *poprawnych złożzeń* elementów z zbioru  $\bigcup(V_i, i \neq d, i \in A)$ , przy czym przez *poprawne złożenie* należy rozumieć takie złożenie, które zawiera maksymalnie jeden element z danego  $V_i$ . Elementy zbiorów są wstępnie odfiltrowane przez zbadanie wsparcia  $sup$ , które liczbowo jest równe liczbie obiektów w tablicy decyzyjnej o wartościach atrybutów jak w  $\rho_i, i = \{1, 2\}$ .

**Przykład 2.5.1** Załóżmy prostą tablicę decyzyjną zaprezentowaną poniżej:

$id$	$a$	$b$	$c$	$d$
$x1$	$a_1$	$b_1$	$c_1$	$d_1$
$x2$	$a_1$	$b_2$	$c_1$	$d_1$
$x3$	$a_2$	$b_2$	$c_1$	$d_2$

W tablicy tej atrybutem decyzyjnym jest atrybut  $d$ , zaś atrybuty  $a, b, c$  są zmiennymi atrybutami warunkowymi. Na tej podstawie możemy ustalić, że:  $V_a = \{a_1, a_2\}$ ,  $V_b = \{b_1, b_2\}$ ,  $V_c = \{c_1\}$ . W takim przypadku poprawnym złożeniem jest  $a_1 \cdot d_1$  lub  $a_1 \cdot b_1 \cdot d_1$ , natomiast  $a_1 \cdot a_2 \cdot d_1$  nie jest poprawnym złożeniem.

W pierwszej iteracji algorytmu ARED tworzony jest dwuelementowy (a więc najkrótszy, jaki może być) zbiór  $\tau$  poprzez złożenie elementów zbioru wartości jednego z atrybutów warunkowych z elementami zbioru wartości atrybutu decyzyjnego. W dalszej części działania algorytmu badane są wszystkie możliwe pary elementów zbiorów  $\tau$  i  $\sigma$ :  $(t, s), t \in \tau, s \in \sigma$ . Postać  $i$ -tej pary to  $(t_i, s_i)$ , a postać każdego

elementu tej pary jest zgodna z podaną wcześniej definicją. Jeśli wartości atrybutu decyzyjnego w ramach danej pary są różne, para ta poddawana jest kilku testom. Jeśli atrybuty warunkowe w każdym elemencie pary są atrybutami stałymi, to elementy tej pary przepisywane są do nowych zbiorów  $\tau$  and  $\sigma$  na potrzeby przyszłych iteracji. Logika stojąca za tą operacją jest następująca: na podstawie definicji zbiorów  $\tau$  i  $\sigma$  wiemy, że co najmniej  $\lambda_1$  obiektów spełnia warunki reprezentowane przez badane złożenie, jednak ze względu na stałość atrybutów nie można zbudować z nich reguły akcji – natomiast w przyszłych iteracjach być może przez dołączenie kolejnego atrybutu warunkowego zmiennego będzie to możliwe.

Jeżeli każdy z elementów badanej pary zawiera różne wartości tego samego atrybutu warunkowego, to konstruuje się kandydującą regułę akcji postaci  $(t_i \rightarrow s_i)$ . Jeżeli  $conf(t_i \rightarrow s_i) \geq \lambda_2$ , to reguła taka dodawana jest do wynikowego zbioru reguł akcji, przeciwnie jej elementy umieszczane są odpowiednio w zbiorach  $\tau$  i  $\sigma$  na potrzeby dalszych iteracji.

Aby wyznaczyć wartość ufności  $conf$  reguły akcji w algorytmie ARED należy najpierw wyznaczyć wartość wsparcia reguły akcji, definiowaną przez autorów następująco:  $sup(ar) = \min(sup(t_i), sup(s_i))$ . Ufność można wyznaczyć z zależności  $conf(ar) = \frac{sup(ar)}{sup(t_1)}$ .

Gdy zakończy się przeglądanie par elementów dwuelementowych zbiorów  $\tau$  i  $\sigma$ , tworzone są ich trójelementowe odpowiedniki (ściśle, tworzone są nowe zbiory składające się z par postaci  $(t_n, d_i)$ , gdzie  $t_n$  jest dwuelementowy). Rozważmy dwa elementy  $\tau_1 = t_1 \cdot d_1$  i  $\tau_2 = t_2 \cdot d_1$  dwuelementowego zbioru  $\tau$ . Przyjmijmy, że  $|t_i|$  oznacza zbiór wartości wszystkich elementów składowych  $t_i$ .  $\tau_1$  i  $\tau_2$  zostaną połączone, gdy  $d_1 = d_2$  (ta sama wartość atrybutu decyzyjnego) oraz  $|t_1 \cup t_2| - |t_1 \cap t_2| = \{v_1 \in V_a, v_2 \in V_b\}$ , gdzie  $a \neq b$  ( $\tau_1$  i  $\tau_2$  mają po jednej wartości ze zbioru wartości różnych atrybutów). Na tej zasadzie powstają nowe zbiory  $\tau$  i  $\sigma$ , które poddawane są opisanej wcześniej procedurze testowej pozwalającej stwierdzić, czy powstała nowa reguła akcji lub odłożyć kandydatów do utworzenia reguł akcji w następnej iteracji.

Proces ten postępuje tak długo, aż nie da się już utworzyć zbiorów  $\tau$  i  $\sigma$ , których elementy można by łączyć w reguły akcji. Metoda ARED pozwala na indukcję reguł bez potrzeby wcześniej indukcji reguł klasyfikacyjnych, operuje jednak tylko na atrybutach nominalnych, co w przypadku obecności atrybutów numerycznych w zbiorze danych wymaga od użytkowników metody przeprowadzenia wcześniejszej dyskretyzacji.

## 2.5.2 Indukcja reguł akcji na podstawie zbiorów reguł decyzyjnych

Kolejna rodzina metod odkrywania reguł akcji wykorzystuje uzyskane wcześniej, przy użyciu innych algorytmów, zbiory reguł decyzyjnych. Idea ta opiera się o obserwację, że aby utworzyć regułę akcji niezbędna jest wiedza na temat klas źródłowej i docelowej. Aby uzyskać tę wiedzę, można zastosować jedną ze znanych metod do scharakteryzowania obserwacji źródłowych i docelowych (np. wyindukować reguły decyzyjne), a następnie połączyć odpowiednio te fakty poprzez sformułowanie reguł akcji. Potencjalnie oszczędza to koszt wstępnej analizy danych, ale także pozwala uzyskać nową wiedzę bazując na już istniejących modelach, a bez konieczności dostępu do danych źródłowych. Przykładem takiej metody jest algorytm DAKAR (ang. Discovery of Actionable Knowledge and Recommendation) [96]. Algorytm DAKAR nie tworzy reguł akcji w sensie opisanym w niniejszym rozdziale. Autorzy tej metody utrzymują, że DAKAR pozwala uzyskiwać rekomendacje, o których szerzej traktuje rozdział 4 niniejszej pracy. Reguły akcji i rekomendacje akcji mogą być reprezentowane w ten sam sposób, jednak nie da się określić rekomendacji bez podania przykładu, dla którego ma ona zostać stworzona. W uproszczeniu można przyjąć, że rekomendacja jest specjalizowaną regułą akcji, wygenerowaną specjalnie dla podanego przykładu. W opisie metody DAKAR używa się stwierdzeń „Sytuacja wejściowa (S)” na określenie przykładu, dla którego szukana będzie rekomendacja (A), która opisuje transformację S do „sytuacji wyjściowej out(S, A)”. Autorzy podają dwie wersje algorytmu, DAKAR1 i DAKAR2. Dalej podano krótki opis działania metody DAKAR2, która jest rozwinięciem i ulepszeniem algorytmu DAKAR1, kładącym większy nacisk na jakość uzyskiwanych rekomendacji.

Danymi niezbędnymi do wykonania algorytmu DAKAR2 są przede wszystkim reguły decyzyjne. Zakładając, że mierzymy się z problemem dwuklasowym, niezbędny będzie zbiór reguł  $R = R^+ \cup R^-$ , gdzie  $R^+(R^-)$  zawiera reguły klasyfikacyjne wskazujące na klasę docelową (źródłową). W ramach metody DAKAR2 przeszukiwana będzie przestrzeń potencjalnych rekomendacji, pod kątem znalezienia takiej, która maksymalizuje kryterium jakości zdefiniowane następująco:  $q(out(S, A)) = \sum_{r \in R^+, r \text{ pokrywa } out(S, A)} prec(r) - \sum_{r \in R^-, r \text{ pokrywa } out(S, A)} prec(r)$ . Nadto autorzy ograniczają przestrzeń możliwych rozwiązań poprzez wyszukiwanie tylko akcji, które dokonują korekcji wartości sytuacji wejściowej, tj. sugerują taką zmianę wartości danego atrybutu, że przecięcie zbioru wartości sytuacji wejściowej i wyjściowej jest



puste (innym rodzajem akcji definiowanym przez autorów DAKAR2 jest specyfikacja - wówczas przecięcie to jest niepuste). Metoda DAKAR2 pozwala na wygenerowanie zadanej liczby rekomendacji, z których co najmniej jedna będzie optymalizować kryterium jakości  $q$ . Kryterium to przyjmuje najwyższą wartość dla danej akcji  $A$  i sytuacji wejściowej  $S$ , gdy sytuacja wyjściowa  $out(S, A)$  jest pokrywana przez wszystkie reguły ze zbioru  $R^+$  i żadną ze zbioru  $R^-$ . DAKAR2 próbuje więc rozwiązać problem znalezienia akcji aplikowalnych i wykonalnych, takich, że zbiór  $R^+$  jest jak największy. Przez akcje aplikowalne autorzy rozumieją takie akcje, które są korektami oraz nie posiadają w przesłance rozłącznych zbiorów dotyczących tego samego atrybutu. Wykonalność akcji definiowana jest dwojako, w zależności od potrzeb użytkownika algorytmu:

- FLEX-wykonalność - akcja uznawana jest za wykonalną, jeśli nie sugeruje modyfikacji żadnego z atrybutów stabilnych
- $\delta$ -wykonalność - opierając się na wagach podanych przez użytkownika oraz odległości Minkowskiego wyliczanej dla wartości atrybutów sytuacji wejściowej i wyjściowej wyznaczany jest niejako „koszt” wykonania akcji jako liczba rzeczywista z zakresu 0-1. Akcja uznawana jest za wykonalną, jeśli wartość ta jest mniejsza od podanego przez użytkownika parametru  $\delta$

Zadanie to jest rozwiązywane przez wyszukiwanie maksymalnej kliku w grafie reguł. Aby zbudować graf umożliwiający takie poszukiwanie, potrzeba zdefiniować reguły połączone. Dwie reguły uznamy za połączone, jeśli zbiory wartości warunków elementarnych w przesłankach tych reguł zdefiniowane dla pewnego atrybutu mają część wspólną. Graf reguł definiowany jest następująco: każda reguła stanowi wierzchołek grafu, a pomiędzy dwoma regułami istnieje krawędź, jeśli reguły te są połączone. Kliką w takim grafie jest oczywiście podgraf, w którym wszystkie reguły są parami połączone. Wyszukując klik w grafie reguł oraz analizując właściwości (aplikowalność, wykonalność) złożzeń reguł budujących taką klikę, algorytm DAKAR2 jest w stanie sugerować akcje oparte zarówno o atrybuty nominalne jak i numeryczne.

### 2.5.3 Generowanie reguł akcji z zastosowaniem strategii wyczerpującej

Opierając się na idei algorytmu Apriori [1], He i in. zaproponowali dwie metody odkrywania reguł akcji [44]. Obydwie metody bazują na strategii zawartej w Apriori

– reguły są stopniowo rozszerzane, a do dalszych etapów przechodzą tylko te, które po rozszerzeniu spełniają zadane przez użytkownika właściwości.

Algorytm MARFS1 (ang. Mining Action Rules From Scratch) generuje reguły akcji spełniające cztery zadane przez użytkownika parametry: minimalne wsparcie pozytywne, minimalne wsparcie negatywne, minimalna ufność i maksymalny koszt. Wsparcie pozytywne (negatywne) jest definiowane jako stosunek przykładów klasy docelowej (źródłowej) pokrywanych przez docelową (źródłową) część reguły do wszystkich przykładów klasy docelowej (źródłowej), z dodatkowym obostrzeniem, że reguła nie pokrywa żadnych przykładów klasy źródłowej (docelowej).

Koszt reguły jest definiowany jako suma kosztów wszystkich akcji w przesłance reguły, przy czym macierz kosztów jest specyficzna dla analizowanego zbioru i zadawana przez użytkownika metody lub eksperta domenowego.

Przez ufność reguły akcji w metodzie MARFS1 należy rozumieć stosunek przykładów klasy źródłowej, które po zaaplikowaniu (przetransformowaniu przykładu zgodnie z przesłanką) reguły są odpowiednikami przykładów klasy docelowej do wsparcia negatywnego reguły (MARFS1 działa tylko dla zbiorów składających się z atrybutów nominalnych).

W opisie metody MARFS1 podkreślono, że regułę akcji dla uproszczenia opisu można traktować jako kolekcję par wartości atrybutu, i właśnie przestrzeń zbudowana z takich obiektów jest przeszukiwana w celu znalezienia „częstych” reguł akcji, będących odpowiednikami zbiorów częstych znanych z Apriori.

Autorzy metody zauważają, że reguły akcji spełniają kryterium „domknięcia z dołu” znane z algorytmu Apriori dla wsparcia pozytywnego i negatywnego oraz kosztu, tj. jeśli pewna reguła ma wartość wsparcia mniejszą od zadanej, to także wszystkie jej podreguły mają wartość wsparcia mniejszą od zadanej. Regułę spełniającą kryterium wsparcia nazywa się regułą częstą. Regułę spełniającą kryterium kosztu, nazywa się regułą o niskim koszcie. Własność ta pozwala odsiewać reguły kandydujące w fazie wzrostu, gdyż nie ma sensu na danym etapie próbować rozszerzać reguły która nie spełnia zadanego przez użytkownika kryterium.

Reguły budowane są iteracyjnie, poprzez złączanie w kroku  $k$  przesłanek reguł krótszych (o długości  $k - 1$ ). Do kolejnych iteracji przechodzą tylko reguły częste i o niskim koszcie, gdyż tylko rozszerzanie takich reguł daje szansę na znalezienie dłuższej reguły częstej i o niskim koszcie. Na koniec każdej reguły sprawdzane jest także kryterium ufności, które nie jest „domknięte z dołu”. Reguły nie spełniające zadanej przez użytkownika minimalnej ufności, są odsiewane ze zbioru reguł o długości  $k$ .

Algorytm kończy się, gdy nie da się już utworzyć żadnych reguł częstych i o niskim koszcie o długości  $k$ . Wynikiem działania są wszystkie reguły, które nie zostały odrzucone w iteracjach  $1, 2, \dots, k - 1$ .

Metoda MARFS2 polega na znalezieniu najpierw zbiorów częstych na przykładach klasy źródłowej oraz docelowej osobno, a następnie łączeniu ich elementów w reguły akcji i testowaniu ich na zgodność z kryteriami podanymi przez użytkownika.

#### 2.5.4 Odkrywanie reguł akcji na podstawie drzew decyzyjnych

Popularną metodą uzyskiwania reguł decyzyjnych jest ich ekstrakcja z wcześniej stworzonych drzew decyzyjnych. Podobną strategię zastosowali Yang i in. [107] do indukcji reguł akcji. Zaprezentowana przez nich metoda nie wytwarza reguł akcji w ujęciu przedstawionym w niniejszej pracy, ale ze względu na łatwość konwersji jej wyników działania do postaci reguł akcji oraz ciekawą zasadę działania algorytmu zostanie on pokrótce opisany. Algorytm *leaf-node search* (LNS) przeszukuje drzewo decyzyjne pod kątem znalezienia zbioru akcji, tj. sugestii zmiany wartości atrybutu, które umożliwią przeniesienie przykładów z mniej pożądanego liścia tego drzewa do bardziej pożądanego. Istotny jest fakt, że liściom drzewa przypisane jest prawdopodobieństwo, że obiekt zaklasyfikowany do tego liścia w drzewie znajduje się w pożądanej klasie. Obowiązuje kilka restrykcji: wszystkie atrybuty analizowanego zbioru danych muszą być symboliczne, a na atrybucie decyzyjnym musi być zdefiniowana relacja porządku (pewne jego wartości są bardziej pożądane od pozostałych). Oprócz tego, ekspert domenowy musi zdefiniować macierz kosztów akcji, która pozwoli spriorytezyzować wybór akcji. Macierz ta nie musi być symetryczna, tj. koszt zmiany pewnego atrybutu z wartości  $A$  na wartość  $B$  może być różny od kosztu zmiany z  $B$  na  $A$ . Autorzy wskazują że istnieją atrybuty, których zmiana jest bardzo kosztowna lub wręcz niemożliwa (np. płeć lub wiek). Takie atrybuty będą określane mianem atrybutów twardych. Atrybutami miękkimi nazwano te atrybuty, których zmiana wartości jest sensowna i charakteryzuje się akceptowalnym kosztem. Jest to rozwiązanie podobne do powszechnie stosowanego w algorytmach indukcji reguł akcji, gdzie mówi się o atrybutach stabilnych (niemodyfikowalnych) i mutowalnych. Metoda LNS została przedstawiona w kontekście problemu maksymalizacji zysków przez firmę ubezpieczeniową poprzez zapewnienie, że klienci pozostaną lojalnymi. Mając ustaloną macierz kosztów akcji, autorzy definiują zysk akcji:  $P_N = P_E - P_{gain} - \sum_i COST_i$   $P_N$  oznacza zysk akcji netto (ang. *Net Profit*),  $P_E$  to zysk z danego klienta, który

posiada pożądaną status (np. lojalny),  $P_{gain}$  oznacza zysk prawdopodobieństwa, że klient przebywa w pożądanym stanie, związany z przeniesieniem go do innego liścia w drzewie, a  $COST_i$  definiuje koszt każdej z akcji niezbędnych do przeniesienia klienta do nowego liścia. Algorytm LNS polega na znalezieniu, dla danego klienta, takiego liścia w drzewie (i towarzyszących akcji), które maksymalizuje wartość  $P_N$ . Dalej autorzy definiują dwie szczegółowe sytuacje, w których zastosowanie może znaleźć algorytm LNS.

Pierwszą z nich jest sytuacja, w której poszukujemy akcji na ograniczonej liczbie atrybutów ( $k$  z  $m$ ,  $k < m$ , gdzie  $m$  to liczba wszystkich atrybutów). Sytuację tę nazywamy problemem ograniczonego zbioru atrybutów (ang. *Bounded Attribute Set Problem*, BAS) W oryginalnej wersji LNS może potencjalnie znaleźć zbiór akcji sugerujący zmiany na każdym atrybucie, co może być sytuacją niepożądaną przez użytkownika metody. Rozwiązaniem problemu jest użycie procedury LNS ze zmodyfikowaną macierzą kosztów. W macierzy, na wszystkich atrybutach oprócz  $k$  wybranych, ustawia się maksymalny koszt, i uruchamia się procedurę LNS. Krok ten powtarza się  $m^k$  razy i ostatecznie wybiera się te  $k$  atrybutów, które pozwoliły zmaksymalizować wartość  $P_N$ . Jest to przeszukiwanie pełne przestrzeni rozwiązań o złożoności  $O(m^k)$ . Autorzy proponują także zachłanny algorytm, w którym stosują schemat wspinaczki: iteracyjnie wybierają atrybut który w danej chwili maksymalizuje wartość  $P_N$ . Wystarczy  $k$  takich iteracji aby znaleźć rozwiązanie, które, jak pokazują autorzy, często jest równoznaczne z tym znalezionym przez wyszukiwanie całkowite.

Autorzy metody LNS podają też problem ograniczonej segmentacji (ang. *Bounded Segmentation Problem*, BSP), który polega na znalezieniu dla zbioru testowego  $k$  grup liści drzewa wskazujących na klasę źródłową a następnie dla każdej z tych grup podania zbioru akcji, które zmaksymalizują wartość  $P_N$  w ramach grupy. Problem BSP rozwiązywany jest podobnym podejściem zachłannym, jak opisany wcześniej BAS. Wynikiem działania algorytmu jest  $k$  zbiorów akcji, wraz z towarzyszącymi im zbiorami przykładów - na podstawie takiej informacji łatwo jest zbudować zbiór reguł akcji, jest to w zasadzie zadanie polegające na zmianie reprezentacji informacji o akcjach.

### 2.5.5 Równoległe odkrywanie reguł akcji

W obliczu coraz to większych składowanych wolumenów danych jak i rosnącej dostępności i szybkości internetu, codziennością staje się konieczność stosowania metod analizy danych, automatycznego odkrywania danych i uczenia maszynowego na masywnych zbiorach danych. Nie inaczej jest w przypadku odkrywania wzorców akcyjnych w postaci reguł akcji. Algorytmy odkrywania reguł są na ogół wymagające obliczeniowo. Skrócenie czasu ich wykonania na dużych zbiorach danych może zostać osiągnięte przez zrównoleglenie obliczeń.

W pracy [98] zastosowano paradygmat Map-Reduce [20] w środowisku Hadoop. Reguły akcji były indukowane w kroku Map na podziałach oryginalnego zbioru danych z użyciem dwóch metod: (1) algorytm LERS + ARoGS [76] i (2) algorytm Association Action Rules [74]. W kroku Reduce wyniki są łączone z zastosowaniem zmodyfikowanego drzewa losowego w trybie głosowania – zachowywane są te reguły akcji, które pojawiły się przynajmniej w połowie kroków Map, a ich wartości pokrycia i ufności są uśredniane. Zachowane reguły są sprawdzane pod kątem spełniania minimalnych wymagań użytkownika co do pokrycia i ufności. Nieodrzucone reguły stanowią wynikowy zbiór reguł akcji. Opisane postępowanie pozwoliło uzyskać zbiory reguł akcji na masywnych zbiorach danych w czasie rzędu kilku minut.

Inne podejście zostało zaprezentowane w pracy [99], w której rozszerzono algorytm MR-Apriori [55] będący metodą indukcji reguł asocjacyjnych w paradygmacie MapReduce. Praca [99] jest adaptacją algorytmu opisanego w [74] w środowisku Hadoop. Podobnie sprawa się ma w przypadku algorytmu SARGs (ang. *Specific Action Rule discovery based on Grabbing Strategy*) [4], który jest rozszerzeniem algorytmu ARoGS [76] i jego adaptacją do równoległego wykonania w środowisku Apache Spark.

### 2.5.6 Zastosowanie reguł akcji

Koncept reguł akcji znalazł już swoje praktyczne zastosowania, które zostały opisane w literaturze, a o których dla kompletności przeglądu należy wspomnieć. Powell i in. [69] pochylił się nad problemem wyceny dzieł sztuki. W swoim badaniu autorzy skolekcjonowali informacje o artystach i ich dziełach sztuki z portalu *Artfinder.com*, który służy do promowania i sprzedaży dzieł sztuki bezpośrednio przez twórców. Na podstawie analizy sentymentu opisu artysty i dzieła sztuki a także cech wizualnych wyekstrahowanych ze zdjęć przy użyciu metod analizy obrazu wyindukowane zostały

reguły akcji, w których atrybutem decyzyjnym była grupa cenowa (10 kategorii cenowych zostało uzyskanych poprzez dyskretyzację ciągłego atrybutu określającego ceny skolekcjonowanych dzieł sztuki). Reguły akcji zostały wytworzone przy użyciu oprogramowania LISP-Miner [110]. Intencją było wskazanie autorom, w jaki sposób powinni prezentować swoje dzieła, aby uzyskać wyższe ceny ich sprzedaży. Analiza została przeprowadzona w podgrupach tak, aby razem analizowane były dzieła o podobnych cechach wizualnych i stylu.

Reguły akcji i meta-akcje zostały wykorzystane przez Tarnowską i in. w konstrukcji rozbudowanego systemu rekomendacyjnego CLIRS (ang. *Customer Loyalty Improvement Recommender System*) [94]. Opisany system został stworzony w ramach firmy z pewnej branży w podniesieniu wyniku NPS (ang. *Net Promoter Score*), poprzez analizę opinii wyrażanych przez klientów w ankietach przeprowadzanych przez firmę konsultingową. Celem autorów systemu było pokazanie działań (meta-akcji), jakie pracownicy firmy powinni podjąć, by zwiększyć wynik NPS. Źródłem danych były ankiety, zawierające zarówno pytania otwarte jak i zamknięte, dotyczące satysfakcji klienta z dostarczanych przez badaną firmę usług. Ponieważ firma konsultingowa obsługiwała ankiety dla wielu klientów instytucjonalnych z branży, system jest w stanie generować rekomendacje ucząc się na danych dotyczących lepiej radzących sobie firm. Aby reguły akcji były indukowane na danych dotyczących podobnych do siebie przedsiębiorstw, autorzy wprowadzili ciekawą miarę semantycznego podobieństwa firm, która opiera się o jakość klasyfikatora regułowego utworzonego w kontekście firmy A, lecz zastosowanego do klasyfikacji ankiet dla klienta B i vice versa. Na podstawie semantycznej odległości między klientami tworzone jest hierarchiczne grupowanie firm, na podstawie którego wybierane są firmy najbardziej podobne do tej, dla której poszukiwane są rekomendacje działań prowadzących do zwiększenia wyniku NPS. Indukcję reguł akcji przeprowadza się na „rozszerzonym” zbiorze danych, który składa się z wyników ankiet specyficznych dla badanego przedsiębiorstwa, jak i przedsiębiorstw najbardziej podobnych semantycznie, ale takich, które uzyskują wyższy wynik NPS. Meta-akcje w systemie CLIRS odkrywane są na podstawie rozbudowanej analizy sentymentu otwartych komentarzy, w której m.in. zestawiane są ze sobą pozytywne i negatywne komentarze dotyczące pewnych cech ocenianej usługi [92]. Druga wersja systemu, CLIRS2 [93] została stworzona w wyniku zmian w formacie przeprowadzanych ankiet polegających na wyrugowaniu większości pytań zamkniętych, a postawieniu na komentarze otwarte. Atrybuty i ich wartości, na podstawie których budowana jest tablica decyzyjna na

której przeprowadzana jest indukcja reguł akcji, są odkrywane automatycznie na podstawie analizy udzielonych komentarzy z zastosowaniem technik NLP. Obydwa systemy, oprócz samej możliwości odkrywania meta-akcji, posiadają rozbudowane możliwości wizualizacji danych i stanowią pełne środowisko pracy dla konsultantów pracujących nad ulepszeniem postrzegania usług klientów instytucjonalnych przez ich odbiorców.

## 3 Indukcja reguł akcji

Poprzedni rozdział przedstawił pokrótce główne spotykane w literaturze strategie odkrywania reguł akcji, które stosowane są już od ponad 20 lat. Indukcja reguł akcji nie jest jednak tematem wyczerpanym - wciąż próbuje się proponować nowe strategie, oparte o mniej utarte w kontekście systemów regułowych podejścia jak programowanie ewolucyjne [42] czy łączenie drzew decyzyjnych i ich reprezentacji regułowej z wiedzą domenową [5].

Niewątpliwie nie opracowano jeszcze metody, którą można uznać za standardowe rozwiązanie gdy pojawia się potrzeba indukcji reguł akcji. Osoba stojąca przed problemem wyboru najlepszej metody odkrywania reguł akcji nie ma łatwego zadania. Wiele metod indukcji reguł akcji zostało opisanych w literaturze, lecz brak jest publicznie dostępnych kodów źródłowych pozwalających na praktyczne wdrożenia czy wiarygodne porównanie wyników generowanych przez te metody. Większość artykułów skupia się na badanym przypadku, prezentując jak metoda wprowadzona przez autorów przyczyniła się do osiągnięcia sukcesu badawczego lub biznesowego. Brakuje także niezależnej od problemu metody oceny reguł akcji, która byłaby konsensusem pozwalającym zestawić ze sobą różne podejścia, by móc wybrać najbardziej efektywne. W przeciwieństwie do reguł decyzyjnych, nie mamy do dyspozycji zestawu miar jakości reguł pozwalających określić ich przydatność. Podobnie trudno jest znaleźć jednolitą definicję określającą kiedy zbiór reguł akcji jest „dobry”, co nie jest problemem gdy chcemy określić skuteczność zbioru reguł decyzyjnych jako klasyfikatora.

Podsumowując, trudno ocenić użyteczność jakiegokolwiek algorytmu reguł akcji w obliczu opisanych wyżej problemów. Dalsza część pracy opisuje dwie metody indukcji reguł akcji zbudowanych wokół dobrze ugruntowanego paradygmatu odkrywania reguł decyzyjnych, wprowadza algorytm pozwalający uzyskiwać rekomendacje (tj. specjalizowane reguły akcji specyficzne dla pojedynczych przykładów) oraz symulować zastosowanie reguł akcji i rekomendacji celem zbadania ich skuteczności. Metoda uzyskiwania rekomendacji oraz sposób oceny zarówno zbiorów reguł akcji jak i rekomendacji są niezależne od zastosowanej metody odkrywania reguł akcji. Dzięki



temu możliwe jest wykorzystanie opisanej w dalszej części pracy metody weryfikacji reguł akcji do porównywania wyników różnych metod, co stanowi istotny krok naprzód w dziedzinie odkrywania reguł akcji.

W niniejszym rozdziale opisano modyfikacje algorytmu sekwencyjnego pokrywania które pozwalają na odkrywanie reguł akcji. Opisano dwie metody, różniące się tym, która klasa - źródłowa, czy docelowa - ma dominujący wkład w wybór atrybutów i warunków elementarnych budujących akcje znajdujące się w przesłankach reguł.

### 3.1 Indukcja reguł akcji z zastosowaniem paradygmatu sekwencyjnego pokrywania

Metoda sekwencyjnego pokrywania pozwala odkrywać zbiory reguł decyzyjnych o dobrej jakości, co ma pozytywny wpływ na czytelność i łatwość zrozumienia wyniku działania przez jego odbiorcę. Dodatkowo reguły indukowane są do momentu, w którym możliwe jest pokrycie wszystkich lub niemal wszystkich przykładów w zbiorze, co przekłada się na rozmiary zbioru reguł.

Dodatkowo paradygmat ten pozwala na sterowanie procesem indukcji reguł z zastosowaniem miar oceny jakości reguł. Umożliwia to dopasowanie postaci wynikowych reguł do szczególnych potrzeb wynikających z charakterystyki rozwiązywanego problemu analizy danych.

Naturalnym wydaje się próba zastosowania tego podejścia również do indukcji reguł akcji. Wedle najlepszej wiedzy autora, pierwsze zastosowanie paradygmatu sekwencyjnego pokrywania w celu indukcji reguł akcji zostało przedstawione przez autora wspólnie z Sikorą i Wróblem w pracy [57]. Opisany algorytm wpasowuje się w ramy ogólnego pokryciowego algorytmu indukcji reguł przedstawionego w rozdziale drugim (Algorytmy 1 i 2), jednak dla ułatwienia opisu jego pseudokod został przedstawiony poniżej.

Tak samo jak ogólny algorytm pokryciowej indukcji reguł decyzyjnych, pokryciowa indukcja reguł akcji składa się z dwóch wewnętrznych procesów sterowanych miarą jakości  $q$ . Procesy te to specjalizacja reguły akcji, w trakcie której rozwijana jest przesłanka reguły, oraz przycinanie reguły akcji. Użyta miara jakości  $q$  może być jedną z reguł używanych do indukcji reguł decyzyjnych – nie wprowadzono specjalnych miar oceny jakości reguł akcji. Przedstawiony w algorytmie 8 pseudokod zakłada, dla uproszczenia opisu, że zbiór przykładów zawiera tylko przykłady klasy źródłowej

---

**Algorytm 8** Pokryciowy algorytm indukcji reguł akcji

---

**Wejście:**  $D(A, C)$  - zbiór przykładów (zbiór atrybutów  $A$  oraz atrybut decyzyjny  $C$ )

**Wejście:**  $q$  - miara oceny jakości reguł klasyfikacyjnych

**Wejście:**  $mincov$  - wartość minimalnego pokrycia zbioru przykładów

**Wejście:**  $s$  - klasa źródłowa,  $s \in V_C$

**Wejście:**  $t$  - klasa docelowa,  $t \in V_C$

**Wyjście:**  $R$  - zbiór reguł akcji

```

1: function GENERUJREGUŁYAKCJI( $D, q, mincov, s, t$ )
2:    $R \leftarrow \emptyset$ 
3:    $S \leftarrow \{d : d \in D \wedge d[C] \equiv s\}$ 
4:    $D_t \leftarrow D \setminus S$ 
5:   while  $D \setminus D_t \neq \emptyset$  do
6:      $D_s \leftarrow S \cap D$ 
7:      $r \leftarrow$  SPECIALIZUJREGUŁĘAKCJI( $D, D_s, s, t, q, mincov$ )
8:      $r \leftarrow$  PRZYTNIJREGUŁĘAKCJI( $r, D, q$ )
9:      $R \leftarrow R \cup r$ 
10:     $D_s \leftarrow$  POKRYCIE( $r, D$ )
11:     $D \leftarrow D \setminus D_s$ 
12:   end while
13:   return  $R$ 
14: end function

```

---

$s$  i klasy docelowej  $t$ . Możliwe jest rozszerzenie metody na zbiory zawierające więcej niż dwie klasy, jednak wymaga to dokładnego określenia oczekiwanych wyników, ponieważ metodę tę można wykorzystać tak samo dobrze do wyindukowania reguł akcji opisujących przejścia między wszystkimi parami klas decyzyjnych jak i do znalezienia w zadanym zbiorze wieloklasowym przejść między kilkoma interesującymi wartościami atrybutu decyzyjnego.

Drobną różnicą względem klasycznej pokryciowej indukcji reguł jest fakt, że w trakcie pokryciowego odkrywania reguł akcji przykłady klasy docelowej nigdy nie są usuwane ze zbioru treningowego, co ma na celu zachowanie pełnej wiedzy o postaci przykładów klasy docelowej na potrzeby procesu wzrostu i przycinania nowych reguł. Przykłady klasy źródłowej pokryte przez nowo-odkrytą regułą akcji są oczywiście usuwane ze zbioru treningowego, co pozwala ewentualnie spełnić warunek głównej pętli *while*. Warunek ten nie musi być tak ostry, jak zaprezentowany w linii 5 algorytmu 8 - zazwyczaj proces indukcji jest kontynuowany aż w zbiorze treningowym pozostaje mały procent oryginalnej liczby przykładów. Ma to na celu ograniczenie występowania reguł bardzo specyficznych w wynikowym zbiorze reguł akcji. W przypadku pokryciowej indukcji reguł akcji pokrycie reguły mierzy się tylko w kontekście przykładów klasy źródłowej. Ta sama zasada stosuje się do warunku głównej pętli *while*.

Metoda specjalizująca regułę akcji, której pseudokod przedstawiono w algorytmie 9, wykorzystuje fakt, że reguła akcji może być traktowana jako złożenie dwóch reguł decyzyjnych. Początkowo tworzona jest reguła akcji o pustej przesłance i z góry zdefiniowanej konkluzji  $s \rightarrow t$ . W celu odkrycia akcji mających budować przesłankę tej reguły utrzymywane są dwie tymczasowe reguły klasyfikacyjne, których konkluzje wskazują odpowiednio na klasę źródłową i docelową.

Dodanie jednej akcji do przesłanki reguły akcji wymaga najpierw znalezienia najlepszego warunku elementarnego opisującego przykłady klasy źródłowej. Odbywa się to w podobny sposób, w jaki specjalizowana jest reguła decyzyjna w algorytmie 3: generowane są wszystkie warunki elementarne, tymczasowa reguła decyzyjna jest rozszerzana o każdy z tych warunków i badana jest jakość tak rozszerzonej reguły. Warunek elementarny dający największy wzrost (lub niepowodujący spadku) jakości reguły jest wybierany jako najlepszy dla klasy źródłowej i dodawany do przesłanki tymczasowej reguły wskazującej na klasę źródłową. Następnie odczytywany jest atrybut, w oparciu o który zbudowany został wybrany najlepszy warunek elementarny i wyszukiwany jest najlepszy warunek elementarny opisujący przykłady klasy

docelowej, z tym że przeszukiwanie odbywa się tylko w zakresie atrybutu, który został odczytany. Dzięki takiemu postępowaniu po pierwsze możliwe będzie utworzenie akcji elementarnej, a po drugie do zbudowania akcji wykorzystywany jest atrybut, który pozwoli pokryć jak najwięcej przykładów klasy źródłowej (lub spełnić inne kryterium jakości, zgodnie z wybranym  $q$ ). Dodatkowo, możliwych jest parę strategii wyboru dopuszczalnych warunków elementarnych opisujących przykłady klasy docelowej. Pierwsza z nich nie nakłada żadnych ograniczeń na postać wybranego warunku elementarnego. Druga strategia zakłada, że docelowego warunku elementarnego szukamy tylko wśród warunków kandydujących, które nie mają części wspólnej z źródłowym warunkiem elementarnym. Trzecia proponowana strategia jest niejako kompromisem pomiędzy tymi dwoma. Zbiór dopuszczalnych, docelowych warunków elementarnych jest w niej budowany w następujący sposób: wyznacza się części wspólne oraz różnice wszystkich kandydujących docelowych warunków elementarnych z wyznaczonym już źródłowym warunkiem elementarnym. Części wspólne odrzuca się, zaś różnice wraz z warunkami elementarnymi nie przecinającymi się z wyznaczonym źródłowym warunkiem elementarnym stanowią zbiór przeszukiwany pod kątem znalezienia najlepszego docelowego warunku elementarnego. Strategie te oczywiście mają zastosowanie do warunków opartych o atrybuty ciągłe.

Przesłanki obu tymczasowych reguł zostają rozszerzone o znalezione najlepsze warunki elementarne, zaś specjalizowana reguła akcji jest rozszerzana o akcję utworzoną z tych warunków elementarnych.

Opisany proces jest powtarzany tak długo, jak długo możliwe jest znalezienie warunku elementarnego dla przykładów klasy źródłowych spełniającego warunek minimalnego pokrycia *mincov*. Z procedury zwracana jest tylko utworzona reguła akcji  $r$ , reguły tymczasowe  $r_t$  i  $r_s$  zostają zapomniane. Nie powoduje to utraty informacji, ponieważ jak opisano to w rozdziale pierwszym, regułę akcji można łatwo sprowadzić do postaci dwóch reguł decyzyjnych i vice versa. Własność ta jest wykorzystywana w opcjonalnym kroku przycinania reguły akcji.

**Przykład 3.1.1** *Prześledźmy wzrost jednej reguły akcji (Algorytm 9) na zbiorze Monk1, syntetycznym zbiorze danych pochodzącym z repozytorium UCI [23]. Wszystkie atrybuty warunkowe  $\{a_1, a_2, a_3, a_4, a_5, a_6\}$  w tym zbiorze są kategoryczne. Atrybut decyzyjny class przyjmuje wartość 1, gdy spełniony jest jeden z dwóch warunków:  $a_1 = a_2$  lub  $a_5 = 1$ . W przeciwnym przypadku, atrybut class przyjmuje wartość 0. Przyjmijmy, że klasa 0 jest klasą źródłową, a 1 – docelową.*

### 3 Indukcja reguł akcji

Początkowo reguła ma pustą przesłankę i ustaloną konkluzję w postaci  $(class, 0 \rightarrow 1)$ . Warunki elementarne wybierane w następujących po sobie iteracjach, które budują akcje elementarne dodawane do przesłanki reguły akcji przedstawione zostały w poniższej tabeli (oznaczenia jak w Algorytmie 9):

iteracja	$w_{best_s}$	$q_{r_s}$	$w_{best_t}$	$q_{r_t}$
1	$(a_1 = 1)$	0.69	$(a_1 = 3)$	0.70
2	$(a_2 = 2)$	0.88	$(a_2 = 3)$	1.00
3	$(a_6 = 2)$	0.90	$(a_6 = 2)$	1.00

**Tabela 3.1:** Kolejne warunki elementarne wybierane w trakcie wzrostu reguły akcji, gdy użytą miarą oceny jakości jest precyzja reguły.

Po zakończeniu specjalizacji reguły, prezentuje się ona następująco:

**IF**  $((a_1 = 1) \rightarrow (a_1 = 3)) \wedge ((a_2 = 2) \rightarrow (a_2 = 3)) \wedge ((a_6 = 2) \rightarrow (a_6 = 2))$  **THEN**  $(class = 0) \rightarrow (class = 1)$ .

Można ją scharakteryzować w kontekście zbioru treningowego jakim był pełny zbiór *Monk1* następującymi wartościami liczbowymi:  $p_s = 9, n_s = 1, p_t = 17, n_t = 0$

Przez przycięcie reguły akcji rozumiemy taką modyfikację przesłanki tej reguły, w wyniku której akcje elementarne są usunięte lub przycięte. Przez przycięcie akcji elementarnej rozumiane jest usunięcie jej konkluzji (jej prawej strony), tj. zredukowanie akcji elementarnej do postaci akcji pustej. Podobnie jak wzrost reguły, proces ten sterowany jest miarą oceny jakości  $q$ . Pseudokod tej operacji został przedstawiony w algorytmie 10.

Proces przycinania rozpoczyna się od rozłożenia reguły akcji na dwie tymczasowe reguły decyzyjne. Przesłanki tych reguł zdeterminowane są przez odpowiednio lewe i prawe strony (przesłanki i konkluzje) akcji elementarnych zawartych w przesłance przycinanej reguły akcji.

Konkluzje tych reguł odpowiadają klasie źródłowej i docelowej oryginalnej reguły akcji. Przycinanie reguł odpowiedzialnych za opis klasy źródłowej i docelowej będzie przebiegać niezależnie, z zastosowaniem dowolnego algorytmu przycinania reguł decyzyjnych. W implementacji zastosowanej do przeprowadzenia badań w ramach niniejszej pracy, proces przycinania reguły decyzyjnej polega na cyklicznym tymczasowym usunięciu warunku elementarnego z przesłanki reguły i sprawdzeniu, czy jakość

---

**Algorytm 9** Specjalizacja reguły akcji

---

**Wejście:**  $D(A, C)$  - zbiór przykładów (zbiór atrybutów  $A$  oraz atrybut decyzyjny  $C$ )

**Wejście:**  $D_s$  - zbiór niepokrytych jeszcze przykładów klasy źródłowej

**Wejście:**  $D_t$  - zbiór wszystkich przykładów klasy docelowej

**Wejście:**  $q$  - miara oceny jakości reguł klasyfikacyjnej

**Wejście:**  $mincov$  - minimalne dopuszczalne pokrycie reguły

**Wejście:**  $s$  - klasa źródłowa

**Wejście:**  $t$  - klasa docelowa

**Wyjście:**  $r$  - reguła akcji

```

1: function SPECJALIZUJREGUŁĘAKCJI( $D, D_s, q, s, t, mincov$ )
2:    $r \leftarrow (\emptyset \rightarrow (s \rightarrow t))$ 
3:    $\triangleright$  Puste reguły decyzyjne wskazujące na klasy źródłową i docelową
4:    $r_s \leftarrow (\emptyset \rightarrow C = s), \quad r_t \leftarrow (\emptyset \rightarrow C = t)$ 
5:   repeat
6:      $w_{best_s} \leftarrow \text{NAJLEPSZYWARUNEKELEMENTARNY}(r_s, D, q, mincov)$ 
7:      $atr \leftarrow \text{ATRYBUT}(w_{best_s})$ 
8:      $w_{best_t} \leftarrow \text{NAJLEPSZYWARUNEKELEMENTARNYDLAATRYBUTU}(atr, r_t, D_t, q, mincov)$ 
9:      $r_s \leftarrow r_s \wedge w_{best_s}, \quad r_t \leftarrow r_t \wedge w_{best_t}$ 
10:     $\triangleright$  Rozszerz regułę akcją zbudowaną ze znalezionych warunków elementar-
        nych
11:     $r \leftarrow r \wedge \text{AKCJA}(w_{best_s}, w_{best_t})$ 
12:  until ( $w_{best_s} = \emptyset$ )
13:  return  $r$ 
14: end function

```

---

tak zmienionej reguły wzrosła lub nie zmieniła się względem sytuacji sprzed usunięcia warunku. Jeśli tak się stało, warunek jest trwale usuwany z przesłanki. Sytuacja, w której jakość przycinanej reguły rośnie jest oczywiście korzystna. Brak zmiany w jakości reguły po usunięciu jednego z warunków elementarnych oznacza, że dany warunek nie wnosi nic do przesłanki, nie ma wpływu na zdolność selekcji przykładów przez tę przesłankę, przez co jest zbędny i może być usunięty. Istnienie takich warunków może być efektem różnej kolejności dodawania warunków elementarnych do przesłanki reguły, na przykład proces indukcji mógł przebiec tak, że najpierw został dodany warunek bardziej szczegółowy, a potem mniej szczegółowy, który pokrywa także wszystkie obserwacje pokryte już przez warunek bardziej szczegółowy. Wówczas bardziej szczegółowy warunek staje się niepotrzebny dla celów klasyfikacyjnych i tylko zaciemnia przesłankę reguły.

Algorytm przycinania reguł opisany w algorytmie 10 wywołuje procedurę przycinania reguł decyzyjnych niezależnie dla reguły źródłowej i docelowej, które powstały z rozkładu przycinanej reguły akcji. Następnie odbywa się rekonstrukcja przesłanki przyciętej reguły akcji, poprzez dodawanie do niej akcji utworzonych z warunków elementarnych przyciętych reguł decyzyjnych. Najpierw przeglądana jest przesłanka przyciętej reguły źródłowej. Dla każdego znajdującego się w niej warunku elementarnego wyszukiwany jest odpowiadający warunek elementarny w przesłance przyciętej reguły docelowej. Przez „odpowiadający” warunek elementarny należy rozumieć warunek elementarny zbudowany na takim samym atrybucie. W dalszym ciągu przeglądana jest przesłanka przyciętej reguły docelowej, w której wyszukuje się takich warunków elementarnych, które nie posiadają swoich odpowiedników w przesłance przyciętej reguły źródłowej. Celem tego działania jest dodanie do przesłanki przyciętej reguły takich akcji, które zawężają przestrzeń cech tylko dla przykładów klasy docelowej.

W wyniku przycinania reguły akcji, każda z akcji elementarnych może zostać przetworzona do jednej z czterech postaci:

- Akcja elementarna postaci  $(A_k, w_s \rightarrow w_t)$  - oznacza to, że dana akcja nie uległa przycięciu, ponieważ warunki elementarne po stronie źródłowej i docelowej są istotne przy zadanej mierze jakości  $q$ ;
- Akcja pusta - powstaje w przypadku gdy warunek elementarny oparty o pewien atrybut  $A_k$  został usunięty z reguły docelowej, ale nie z reguły źródłowej.

Wskazuje to, że atrybut  $A_k$  może być istotny dla selekcji przykładów źródłowej, ale nie jest ważny dla ograniczenia przestrzeni w klasie docelowej;

- Akcja zawężająca - wskazuje na to, że dany atrybut jest istotny dla strony docelowej, ale niekoniecznie dla strony źródłowej. Aby uzyskać ten rodzaj akcji, niezbędna jest druga pętla w algorytmie 10
- Brak akcji - jeśli warunki elementarne oparte o dany atrybut zostaną usunięte zarówno z przesłanki reguły źródłowej jak i docelowej, to w wynikowej przyciętej regule akcji dana akcja elementarna nie pojawi się w ogóle.

---

**Algorytm 10** Przycinanie reguły akcji

---

**Wejście:**  $D(A, C)$  - zbiór przykładów (zbiór atrybutów  $A$  oraz atrybut decyzyjny  $C$ )

**Wejście:**  $r$  - przycinana reguła akcji

**Wejście:**  $q$  - miara oceny jakości reguł

**Wyjście:**  $r'$  - przycięta reguła akcji

```

1: function PRZYTNIJREGUŁĘAKCJI( $D, r, q$ )
2:    $r' \leftarrow \emptyset \rightarrow (s \rightarrow t)$ 
3:    $\triangleright$  Rozkład reguły akcji na reguły decyzyjne
4:    $r_s \leftarrow \text{REGUŁAŹRÓDŁOWA}(r)$ ,  $r_t \leftarrow \text{REGUŁADOCELOWA}(r)$ 
5:    $r'_s \leftarrow \text{PRZYTNIJ}(r_s, D, q)$ 
6:    $r'_t \leftarrow \text{PRZYTNIJ}(r_t, D, q)$ 
7:   for  $w \in \text{PRZESŁANKA}(r'_s)$  do
8:      $w_t \leftarrow \{w' \in \text{PRZESŁANKA}(r'_t) : \text{ATRYBUT}(w') = \text{ATRYBUT}(w)\}$ 
9:      $r' \leftarrow r' \wedge \text{AKCJA}(w, w_t)$ 
10:  end for
11:  for  $w \in \text{PRZESŁANKA}(r'_t)$  do
12:     $w_s \leftarrow \{w' \in \text{PRZESŁANKA}(r'_s) : \text{ATRYBUT}(w') = \text{ATRYBUT}(w)\}$ 
13:    if  $w_s = \emptyset$  then
14:       $r' \leftarrow r' \wedge \text{AKCJA}(\emptyset, w)$ 
15:    end if
16:  end for
17:  return  $r'$ 
18: end function

```

---

**Przykład 3.1.2** *Kontynuując przykład 3.1.1, po zakończonej fazie wzrostu rozpoczyna się przycinanie reguły. Kolejne kroki przycinania przedstawione zostały w tabeli*



### 3 Indukcja reguł akcji

przesłanka	$q_{r_s}$	$q_{r_t}$
$(a_1 = 1) \rightarrow (a_1 = 3) \wedge (a_2 = 2) \rightarrow (a_2 = 3) \wedge (a_6 = 2) \rightarrow (a_6 = 2)$	0.13	0.27
$(a_1 = 1) \rightarrow (a_1 = 3) \wedge ((a_2 = 2) \rightarrow (a_2 = 3)) \wedge (a_6 = 2) \rightarrow$ $(a_1 = 1) \rightarrow (a_1 = 3) \wedge (a_2 = 2) \rightarrow (a_2 = 3)$	0.13	0.27
$(a_1 = 1) \rightarrow (a_1 = 3) \wedge ((a_2 = 2) \rightarrow (a_2 = 3))$	0.21	0.27
$(a_1 = 1) \rightarrow (a_1 = 3) \wedge \rightarrow (a_2 = 3)$	0.21	0.24
$(a_1 = 1) \rightarrow (a_1 = 3) \wedge \rightarrow (a_2 = 3)$	0.26	0.27

**Tabela 3.2:** Ilustracja przycinania reguły z zastosowaniem miary RSS.

3.2, która prezentuje kolejne uzyskane przesłanki reguły akcji, oraz wyznaczoną jakość reguł składowych  $r_s$  i  $r_t$ . W procesie przycinania użyto miary RSS.

Na koniec przycinania, reguła przyjmuje następującą formę:

**IF**  $((a_1 = 1) \rightarrow (a_1 = 3)) \wedge (\rightarrow (a_2 = 3))$  **THEN**  $(class, 0) \rightarrow (class, 1)$

Przycięta reguła charakteryzuje się następującymi wartościami liczbowymi:  $p_s = 31, n_s = 14, p_t = 17, n_t = 0$

Ostatecznie przycięta reguła jest dołączana do wynikowego zbioru reguł akcji, a wszystkie przykłady, które są pokryte przez jej lewą część (tzn. regułę decyzyjną zbudowaną z przesłanek akcji elementarnych zawartych w przesłance tej reguły akcji) zostają usunięte ze zbioru treningowego. Proces specjalizacji (wzrostu) i przycinania nowych reguł jest kontynuowany tak długo, jak tylko dostępne są jeszcze przykłady klasy źródłowej (lub jest ich wystarczająco wiele) w zbiorze treningowym.

## 3.2 Wpływ kierunku indukcji na postać reguł akcji

Algorytm indukcji reguł akcji przedstawiony w poprzedniej sekcji jest sterowany miarą jakości  $q$ , ale duży wpływ na jego wyniki ma charakterystyka analizowanego zbioru danych. W szczególności, dominującą rolę przejmują przykłady klasy źródłowej. Widać to w linii 6 algorytmu 9, gdzie dokonywany jest *de facto* wybór atrybutu, w oparciu o który będzie budowana kandydująca akcja elementarna w trakcie wzrostu przesłanki reguły.

Ideą przyświecającą takiemu postępowaniu, tj. wyborowi atrybutu, który najlepiej w danej sytuacji opisuje przykłady klasy źródłowej jest zapewnienie jak najlepszej jakości reguły źródłowej, a następnie wyszukiwanie najlepszych warunków elementarnych

opisujących przykłady klasy docelowej. Innymi słowy, budowane akcje mają pomóc szybko przenieść przykład z klasy źródłowej - jeśli budowanie akcji rozpoczniemy od klasy źródłowej przy użyciu odpowiedniego warunku elementarnego, a następnie znajdziemy warunek elementarny oparty o ten sam atrybut który najlepiej wydziela przykłady klasy docelowej, mamy szansę zbudować takie akcje, które będą mieć duży wpływ na przykłady klasy źródłowej (rozumiany jako np. wysokie wsparcie reguły w klasie źródłowej, lub zdefiniowaną inaczej jakość reguły).

Postępowanie to intuicyjnie jest słuszne, jednak obarczone wadą, która szczególnie mocno objawia się w zbiorach niezbalansowanych, w których przykładów klasy źródłowej jest znacznie więcej niż docelowej. Opisany dotychczas proces indukcji reguł akcji faworyzuje atrybuty, które pozwalają dobrze opisać klasą źródłową, nie biorąc pod uwagę klasy docelowej. Może to prowadzić do wygenerowania akcji mających wysoką wartość wsparcia w klasie źródłowej, ale znikomą w klasie docelowej. Akcje takie dobrze opisują, którego atrybutu wartość należy zmienić, ale nie są w stanie wskazać wartości popartej wiedzą o charakterystyce klasy docelowej, do której należy go doprowadzić, aby sprowokować zmianę klasyfikacji przykładów. Nie możemy także mieć pewności, że atrybut który dobrze opisuje klasę źródłową, równie dobrze opisuje przykłady klasy docelowej.

Rozważania te doprowadziły do zaproponowanej przez autora wraz Sikorą i Wróblem modyfikacji opisanej w pierwszej sekcji metody w taki sposób, aby indukcja reguł uwzględniała także wiedzę o przykładach klasy docelowej [58].

Modyfikacja ta pozwala na zmianę kierunku indukcji reguł akcji, przez co rozumie się wyszukiwanie nie tylko reguł o przejściu między klasami zadany przez użytkownika, ale także reguł odwrotnych, w konkluzji których klasy zamienione są miejscami, tzn. przeciwnie do intencji użytkownika, zamiast wyszukiwać reguł opisujących przejście z klasy źródłowej do docelowej, szuka się także reguł dla przejść z klasy docelowej do źródłowej. Ostatecznie reguły takie sprowadzane są do postaci zgodnej z zadaną przez użytkownika oraz dołączane do wynikowe zbioru reguł, który jest w efekcie jest obszerniejszy niż uzyskany przy użyciu metody opisanej w poprzedniej sekcji, ale zawarte w nim reguły akcji mogą uwzględniać subtelne zależności, których odkrycie nie jest możliwe przy indukcji w jednym kierunku. Oznacza to także, że tracona jest cecha zbiorów reguł uzyskanych przez metody sekwencyjnego pokrywania, która mówi o tym, że uzyskuje się tyle reguł ile wystarczy do pokrycia przykładów w zbiorze. Reguły indukowane zgodnie z intencją użytkownika, z klasy docelowej do źródłowej, nazywane będą regułami indukowanymi wprost lub w przód.

Indukcja reguł w odwrotnym kierunku, z klasy docelowej do źródłowej, nazywana będzie indukcją wstecz, lub indukcją odwrotną.

---

**Algorytm 11** Dwukierunkowa indukcja reguł akcji
 

---

**Wejście:**  $D(A, C)$  - zbiór przykładów (zbiór atrybutów  $A$  oraz atrybut decyzyjny  $C$ )

**Wejście:**  $q$  - miara oceny jakości reguł klasyfikacyjnych

**Wejście:**  $mincov$  - wartość minimalnego pokrycia zbioru przykładów

**Wejście:**  $s$  - klasa źródłowa,  $s \in V_C$

**Wejście:**  $t$  - klasa docelowa,  $t \in V_C$

**Wyjście:**  $R$  - zbiór reguł akcji

```

1: function DWUKIERUNKOWOGENERUJREGUŁYAKCJI( $D, q, mincov, s, t$ )
2:    $R \leftarrow \emptyset$ 
3:    $F \leftarrow$ GENERUJREGUŁYAKCJI( $D, q, mincov, s, t$ )
4:    $B \leftarrow$ GENERUJREGUŁYAKCJI( $D, q, mincov, t, s$ )
5:   for  $rule \in B$  do
6:      $rule \leftarrow$ ODWRÓĆREGUŁĘAKCJI( $rule$ )
7:     ZMIERZPOKRYCIE( $rule, D$ )
8:   end for
9:    $R \leftarrow F \cup B$  return  $R$ 
10: end function

```

---

Pseudokod algorytmu 11 opisuje sposób dwukierunkowej indukcji reguł akcji. Dla uproszczenia opisu wykorzystano w nim odwołanie do funkcji *GenerujRegułyAkcji* opisanej wcześniej w algorytmie 8. W rzeczywistości metodę tę można zaimplementować w taki sposób, aby minimalizować ilość przebiegów po wszystkich przykładach zbioru treningowego, co pozwala na skrócenie czasu obliczeń ale w znacznym stopniu komplikuje opis metody. Ostatecznie oba podejścia dają takie same wyniki, stąd na potrzeby opisu podano algorytm prostszy.

W celu uwzględnienia informacji możliwej do odkrycia przez analizę najpierw przykładów klasy docelowej generuje się dwa zbiory reguł: zbiór  $F$  zawierający reguły akcji o konkluzjach zgodnych z intencją użytkownika metody, oraz zbiór  $B$  zawierający reguły akcji, których konkluzja jest przeciwna do zadanej przez użytkownika. Nazwy zbiorów to pierwsze litery ich angielskich nazw: *Forward Action Rules*, *Backward Action Rules*. W języku polskim nazywane są zbiorem reguł akcji wprost ( $F$ ) i zbiorem odwrotnych reguł akcji, lub reguł akcji wstecz ( $B$ ). „Kierunek” zbioru reguł akcji

jak i samych reguł akcji jest zawsze ustalany w kontekście przejścia zadanego przez użytkownika metody.

Zanim reguły zgromadzone w  $B$  zostaną dołączone do wynikowego zbioru reguł, muszą zostać przetworzone tak, aby opisywać przejście z klasy źródłowej do docelowej. Z tego powodu wszystkie reguły w  $B$  są odwracane - procedurę tę obrazuje algorytm 12. Po sprowadzeniu reguł do pożądanej postaci, trzeba ponownie zmierzyć ich pokrycie, aby użytkownik finalnych reguł był w stanie ocenić ich rzetelność. Konieczność ta wynika z faktu dezaktualizacji informacji o pokryciu w trakcie odwracania reguł.

Odwracanie reguł wymaga przede wszystkim zamiany kolejności klas w konkluzji reguły. Po odwróceniu reguły akcji, klasa źródłowa stanie się docelową i vice versa. W dalszym ciągu konieczne jest także zmodyfikowanie przesłanki reguły akcji. Każda z akcji elementarnych zostaje poddana prostej analizie. Jeśli mamy do czynienia z pełną akcją elementarną, zamienia się miejscami wartości (przesłankę i konkluzję) tej akcji. Przypadkiem szczególnym jest akcja, która w wyniku wcześniejszej indukcji lub przycinania nieodwróconej reguły akcji została sprowadzona do akcji pustej (warunku elementarnego). Aby odwrócić taką akcję, wprowadza się specjalny symbol  $ANY$ . Wynik następującego testu: „czy pewna wartość atrybutu zadanego przykładu zawiera się w zakresie warunku elementarnego opisanego symbolem  $ANY$ ” jest zawsze prawdziwy. W wyniku odwrócenia warunku elementarnego (zredukowanej akcji elementarnej) postaci  $(A_k, v_l)$  otrzymujemy akcję elementarną  $(A_k, ANY \rightarrow v_l)$ , którą należy interpretować jako sugestię zmiany wartości atrybutu  $A_k$  badanego przykładu do wartości  $v_l$  niezależnie od jego bieżącej wartości. Proces odwracania reguły akcji całkowicie zmienia jej postać i sens, co pociąga późniejszą konieczność aktualizacji informacji o pokryciu odwróconej reguły. Należy zwrócić uwagę na to, że akcja typu  $ANY$  może powstać tylko z akcji pustej w oryginalnej regule. Akcja zawężająca w regule oryginalnej w wyniku odwrócenia zostanie przekształcona do postaci akcji pustej, równoważnej warunkowi elementarnemu. Akcja podtrzymująca jest oczywiście taka sama przed, jak i po odwróceniu.

W trakcie tworzenia zbioru  $B$  uzyskiwana jest informacja o atrybutach najbardziej znaczących dla klasy docelowej, oraz o wartościach, które najlepiej opisują przykłady klasy źródłowej i docelowej w ramach tego atrybutu. Odwracanie reguł wyindukowanych wstecz jest niezbędnym procesem prowadzącym do utworzenia pożądanych sugestii zmian wartości atrybutów.

Indukcja reguł wstecz i ich odwracanie pozwala na odkrycie subtelnych zależności, których znalezienie przy prowadzeniu indukcji wprost nie byłoby możliwe z powodu

---

**Algorytm 12** Odwracanie reguł akcji

---

**Wejście:**  $r$  - reguła akcji**Wyjście:**  $r'$  - odwrócona reguła akcji

```

1: function ODWRÓCREGUŁĘAKCJI( $r$ )
2:    $S \leftarrow$  KLASAZRÓDŁOWA( $r$ )
3:    $T \leftarrow$  KLASADOCELOWA( $r$ )
4:    $r' \leftarrow \emptyset \rightarrow (T \rightarrow S)$ 
5:   for  $akcja \in$  PRZESŁANKA( $r$ ) do
6:     if  $akcja$  jest warunkiem elementarnym then
7:        $r' \leftarrow r' \cup$  AKCJA(ANY, Wartość( $akcja$ ) )
8:     else
9:        $r' \leftarrow r' \cup$  AKCJA(Konkluzja( $akcja$ ), Przesłanka( $akcja$ ))
10:    end if
11:  end for
12: end function

```

---

niedopuszczenia pewnych warunków elementarnych ze względu na zastosowaną miarę jakości  $q$  oraz ograniczenie minimalnego pokrycia przesłanki tworzonej reguły *mincov*. Ograniczenie to może powodować odrzucenie warunków elementarnych, które charakteryzują się dużym pokryciem w klasie docelowej, ale niewielkim w klasie źródłowej, nawet jeśli pokrycie to pozwala uzyskać wysoką wartość miary  $q$ . Połączenie reguł wyindukowanych wprost z odwróconymi regułami wyindukowanymi wstecz pozwala na uwzględnienie warunków elementarnych, które normalnie byłyby odrzucone.

Dodatkową modyfikacją algorytmu 11 pozwalającą na uzyskanie bardziej zwięzłego zbioru reguł jest filtracja odwróconych reguł ze zbioru  $B$ . Po zmierzeniu pokrycia odwróconych reguł na zbiorze treningowym znane są wszystkie zmienne niezbędne do wyliczenia wartości  $q$  dla tych reguł. Użytkownik metody może zadać progową wartość jakości odwróconych reguł, aby w wynikowym zbiorze uwzględnić tylko reguły cechujące się wysoką jakością w kontekście przejścia między klasami pożądanego przez użytkownika.

Możliwą modyfikacją algorytmu 11 jest zdanie się na indukcję tylko reguł wprost lub wstecz na podstawie wcześniejszej analizy zbioru treningowego. Na przykład po stwierdzeniu, że zbiór cechuje się silną nadreprezentacją przykładów klasy źródłowej, użytkownik może zdecydować się na indukcję tylko reguł wstecznych. Innym przypad-

kiem, gdy reguły odwrotne mogą być preferowane jest stwierdzenie, że klasy źródłowa i docelowa są dobrze opisywane przez różne zestawy cech. Ze względu jednak na sposób działania algorytmu, nie ma gwarancji, że uzyskany zbiór reguł odwrotnych po odwróceniu do postaci spełniającej założenia użytkownika będzie wystarczający do pokrycia wszystkich przykładów treningowych. Podobnie jednak wybierając tylko indukcję reguł wprost uzyskamy zwężły zbiór reguł akcji, który może cechować się słabą zdolnością do opisu przykładów klasy docelowej. Zaproponowany wyżej algorytm dwukierunkowej indukcji z filtracją wydaje się być dobrym kompromisem pomiędzy liczbą wynikowych reguł akcji a jakością całego zbioru reguł.

W rozdziale 5 przedstawiono wyniki działania pokryciowego algorytmu indukcji reguł akcji w trzech głównych wariantach: tylko indukcja reguł w przód ( $F$ ), tylko indukcja reguł wstecz z odwracaniem reguł akcji ( $B$ ), oraz wariant przedstawiony na algorytmie 11 nazywany fuzją wariantów  $F$  i  $B$ .

## 3.3 Inne rodzaje reguł akcji

Do tej pory niniejsze opracowanie skupiało się na opisie uzyskiwania klasyfikacyjnych reguł akcji. Konkluzja takich reguł akcji zawiera akcję elementarną wskazującą pożądaną zmianę klas. Nie da się ukryć, że reguły akcji opisane jak we wcześniejszych rozdziałach wiele czerpią z koncepcji reguł decyzyjnych. Systemy regułowe znajdują także zastosowanie w zadaniach regresji [100], [52] oraz przeżycia [104]. Przedstawione już metody indukcji reguł akcji można rozszerzyć celem uzyskania odpowiednio reguł akcji regresji oraz reguł akcji przeżycia, o czym traktuje ten podrozdział.

### 3.3.1 Reguły akcji w zadaniu regresji

Przez problem regresji rozumie się przewidywanie wartości pewnego wyróżnionego atrybutu numerycznego (zmiennej zależnej) na podstawie wartości pozostałych atrybutów w danym zbiorze danych (zmiennych niezależnych). W ujęciu systemów reguł, miejsce dyskretnego atrybutu decyzyjnego w konkluzji reguły zajmie atrybut ciągły, zaś przesłanka reguły ma taką samą postać jak w przypadku reguł decyzyjnych. Konkluzja reguły regresji nie wskazuje więc na przynależność przykładu spełniającego warunki przesłanki do pewnej klasy, ale przypisuje mu konkretną wartość atrybutu decyzyjnego.

Inaczej definiuje się również zbiór przykładów negatywnych i pozytywnych dla takiej reguły [48].

**Definicja 7** *Dla danej reguły regresji IF  $w_1 \wedge w_2 \wedge \dots \wedge w_n$  THEN  $d = v$  i pewnych wartości stałych  $\epsilon_1, \epsilon_2$  za zbiór przykładów pozytywnych przyjmuje się wszystkie przykłady ze zbioru treningowego spełniające przesłankę tej reguły o wartości atrybutu decyzyjnego  $d$  wpadającej w zakres  $[v - \epsilon_1, v + \epsilon_2]$ . Wszystkie pozostałe przykłady tworzą zbiór przykładów negatywnych tej reguły. Liczebność tych zbiorów będzie oznaczana odpowiednio  $p_r$  i  $n_r$ . Konsekwentnie, liczebność wszystkich przykładów pozytywnych i negatywnych będzie wyznaczana następująco:*

$$P_r = \sum_{i=1}^n (y_i \in [v - \epsilon_1, v + \epsilon_2]), N_r = n - P_r \quad (3.1)$$

gdzie  $n$  oznacza rozmiar zbioru treningowego,  $y_i$  wartość atrybutu decyzyjnego  $i$ -tego przykładu.

Indukcja regresyjnych reguł akcji pomaga uzyskać odpowiedź na pytanie, w jaki sposób wpływać na obiekty reprezentowane przez przykłady w zbiorze danych, aby uzyskać pożądane wartości atrybutu decyzyjnego. W przeciwieństwie do klasycznych reguł akcji nie jesteśmy jednak w stanie wyróżnić klasy źródłowej i docelowej. Możliwa jest redukcja zadania regresji do zadania klasyfikacji, poprzez zdyskretyzowanie atrybutu decyzyjnego. Wówczas każda z wartości dyskretnego już atrybutu jest skojarzona z pewnym zakresem wartości atrybutu ciągłego. Tę samą redukcję można przeprowadzić w przypadku reguł akcji. Wystarczy wtedy określić, który zakres jest pożądany i przeprowadzić indukcję reguł akcji. Niestety, tak uzyskane reguły nie będą wskazywać konkretnej wartości atrybutu decyzyjnego, potrzebna będzie jeszcze metoda wyboru ostatecznej wartości z zakresu wskazanego przez regułę, co sprawia że takie postępowanie nie zawsze będzie wystarczające.

Nic nie stoi jednak na przeszkodzie wygenerowania reguł akcji, w których konkluzjach znajdzie się wartość atrybutu numerycznego. Akcja elementarna w następniku takiej reguły akcji będzie wskazywać, jak może zmienić się wartość atrybutu decyzyjnego, używając konkretnych wartości tego atrybutu wraz z podaniem stałych  $\epsilon_1, \epsilon_2$ . W niektórych zastosowaniach pożądane będzie, by docelowa część reguły wskazywała przykłady o wyższej wartości atrybutu decyzyjnego (np. zwiększenie zysku, jaki firmie przynosi dany typ klienta), w innych - by była to wartość niższa niż wskazywana przez źródłową część reguły. Możliwa jest także sytuacja, że obie opisane opcje są pożądane, gdy potrzebna jest interwencja, która pozwoli unikać pewnych wartości

atrybutu decyzyjnego. W rozważaniach tych należy jeszcze wspomnieć, że porównując ze sobą dwie regresyjne reguły akcji pod uwagę można wziąć nie tylko wskazywaną przez nie wartość atrybutu decyzyjnego, ale także rozkład wartości tego atrybutu wśród przykładów pokrywanych przez ich przesłanki. W tej pracy nie będziemy się zajmować dalej tym ostatnim zagadnieniem.

W przypadku reguł regresyjnych zbiory przykładów pozytywnych i negatywnych zmieniają się wraz z rozszerzaniem (lub przycinaniem) przesłanki reguły. W przyjętej przez autora implementacji, konkluzja reguły regresji jest determinowana przez jej przesłankę. Wynika to z przyjęcia założenia  $\epsilon_1 = \epsilon_2 = \epsilon$  i ustalenia wartości  $\epsilon$  jako odchylenia standardowego wartości atrybutu decyzyjnego odczytanych z przykładów pokrywanych przez przesłankę reguły. Ta samą zasadę przyjęto przy konstrukcji algorytmu wzrostu reguły akcji regresji przedstawionego w algorytmie 13.

Podobnie jak w przypadku klasycznych reguł akcji, wzrost reguły jest realizowany poprzez utrzymywanie i rozbudowywanie dwóch reguł składowych, z których jedna odpowiedzialna jest za opis przykładów źródłowych, a druga docelowych. Przesłanka reguły źródłowej jest rozszerzana w procesie sterowanym miarą jakości reguł klasyfikacyjnych, tak samo jak w przypadku reguł klasyfikacyjnych (linia 7 algorytmu 13). Po znalezieniu najlepszego warunku elementarnego rozpoczyna się poszukiwanie przeciwstawnego warunku, który posłuży później do skonstruowania akcji rozszerzającej przesłankę budowanej reguły akcji. Poszukiwanie odbywa się tylko w przestrzeni atrybutu, który został wykorzystany do konstrukcji znalezionej już warunku elementarnego, którym rozszerza się regułę źródłową.

Wszystkie znalezione warunki elementarne na danym atrybucie testuje się pod kątem przydatności do rozszerzenia reguły docelowej. W tym celu tymczasowo rozszerza się przesłankę reguły docelowej każdym z kandydujących warunków. Każdorazowe rozszerzenie reguły docelowej powoduje zmianę jej konkluzji. Testowane jest przecięcie zakresów wartości opisanych w konkluzji reguły docelowej i źródłowej (linia 13). Do dalszego przetwarzania wybierane są tylko takie warunki elementarne, dla których przecięcie tych zakresów jest puste, co ma na celu zagwarantowanie, że reguły źródłowa i docelowa są niesprzeczne. W dalszym ciągu badana jest relacja między wartościami konkluzji tymczasowych reguł. Jeśli relacja ta nie jest spełniona, bieżący kandydujący warunek elementarny nie będzie brany pod uwagę przy ostatecznym rozszerzeniu reguły docelowej. Co do zasady, relacja między wartościami konkluzji użyta w linii 16 algorytmu 13 może być dowolnie zadana przez użytkownika, który nie musi ograniczać się do relacji "większy", "mniejszy" czy "różny".



$a$	$d$
3	1
4.5	3
5	5
7	6
8	10

Tabela 3.3: Przykładowa tablica decyzyjna

**Przykład 3.3.1** *Przyjmijmy następującą syntetyczną tablicę decyzyjną z jednym atrybutem warunkowym  $a$  oraz ciągłym atrybutem decyzyjnym  $d$ .*

*Przypuśćmy, że w trakcie budowania regresyjnej reguły akcji, jako najlepszy warunek rozszerzający regułę źródłową wybrano  $a \leq 6$ . Wobec tego, reguła źródłowa przyjmie postać **IF**  $a \leq 6$  **THEN**  $d = 3 \pm 2$ . Aby zbudować regułę akcji, algorytm rozpocznie teraz poszukiwanie warunku rozszerzającego przesłankę reguły docelowej. Przyjmijmy, że testowane są dwa warunki:  $a \geq 5$  oraz  $a \geq 6$ . Jeśli wybrany zostanie pierwszy z nich, reguła docelowa uzyska postać **IF**  $a \geq 5$  **THEN**  $d = 6 \pm 2.6$ , ale zakresy wartości w konkluzji reguły źródłowej i docelowej będą się przecinać, więc warunek ten zostanie odrzucony. W przypadku drugiego warunku, reguła docelowa przyjmie postać **IF**  $a \geq 6$  **THEN**  $d = 8 \pm 2.8$  i jej konkluzja nie będzie się przecinać z konkluzją reguły docelowej.*

*W przyjętej implementacji reguł regresji, wartość konkluzji jest równa medianie wartości atrybutu decyzyjnego odczytanego z przykładów, które zostały pokryte przez przesłankę reguły, a  $\epsilon$  jest równe odchyleniu standardowemu tych wartości.*

Warunki elementarne które spełniają opisane wyżej kryteria są testowane pod kątem jakości, jaką będzie charakteryzować się rozszerzona nimi reguła docelowa. Do zmierzenia jakości wykorzystuje się klasyfikacyjną miarę jakości  $q$ , tę samą, która była już wykorzystana przy wyborze warunku elementarnego rozszerzającego regułę źródłową. Warunek dający w rezultacie najwyższą jakość, jest ostatecznie wybierany by trwale rozszerzyć regułę docelową. Oba najlepsze warunki – źródłowy i docelowy – służą do sformułowania akcji elementarnej, która rozszerzy przesłankę budowanej regresyjnej reguły akcji.

Przedstawiony algorytm dąży do skonstruowania reguły akcji, która będzie wskazywać w zbiorze treningowym na dwie grupy przykładów różniące się wartością atrybutu decyzyjnego. Pozostając w konwencji nazewnictwa reguł akcji, jedna grupa będzie

nazywana źródłową, a druga docelową. Sam proces wzrostu reguły w najprostszym wariacie nie gwarantuje, że grupy te będą się od siebie różnić istotnie. Każda z utworzonych reguł może zostać dodatkowo oceniona pod kątem tego, jak istotnie różne grupy źródłowe i docelowe wskazuje, tj. czy rozkłady wartości atrybutu decyzyjnego budowane przez przykłady z grupy źródłowej i docelowej są statystycznie istotnie różne. Wystarczy dla zadanego poziomu istotności przeprowadzić np. statystyczny test t-Studenta, w którym jako badane próby losowe przyjmujemy odpowiednio wartości atrybutu decyzyjnego przykładów źródłowych i docelowych. Wówczas każda z reguł może zostać oznaczona, jako istotna lub nie, co pomoże w interpretacji ich znaczenia.

Alternatywą jest większe skomplikowanie algorytmu i nałożenie dodatkowych warunków przy wyborze warunków elementarnych rozszerzających przesłanki reguł. Jedną z możliwych modyfikacji jest zastąpienie miary jakości  $q$  (Algorytm 13, linia 19) funkcją maksymalizującą odległość między konkluzją źródłową i docelową.

Przycinanie regresyjnych reguł akcji jest procesem bardzo zbliżonym do opisanego w algorytmie 10. Nakłada się jedynie dwa dodatkowe ograniczenia. Po pierwsze, w sytuacji w której usunięcie tylko części źródłowej lub docelowej którejś z akcji elementarnych może prowadzić do wzrostu lub utrzymania jakości reguły, testowane jest czy w wyniku takiego przycięcia akcji elementarnej nie spowoduje się sytuacji, w której konkluzja reguły źródłowej będzie mieć część wspólną z konkluzją reguły docelowej. Jeśli nie, to takie przycinanie może zajść, przeciwnie – odrzuca się taką próbę optymalizacji przesłanki. Drugim ograniczeniem jest kontrola, czy przycięcie przesłanki nie spowoduje, że przestanie być spełniana zadana przez użytkownika relacja między wartościami konkluzji (np. czy przycięcie nie spowoduje, że przesłanka zarówno źródłowej jak i docelowej reguły będą wskazywać tę samą wartość).

W zastosowaniach praktycznych test na rozłączność zakresów konkluzji ((Algorytm 13, linia 13) oraz test na relację między wartościami konkluzji (Algorytm 13, linia 16) są opcjonalne. Ich zastosowanie wprowadza silne restrykcje na dopuszczalne postaci reguł, co często powoduje szybkie zakończenie sekwencyjnego pokrywania ze względu na niemożność znalezienia nowych warunków elementarnych dających reguły spełniające te obostrzenia. Wyłączenie tych sprawdzeń upraszcza zarówno proces wzrostu jak i przycinania reguły.

### 3.3.2 Reguły akcji w zadaniu analizy przeżycia

Analiza przeżycia znajduje szerokie zastosowanie w medycynie w zadaniu przewidywania czasu wystąpienia wydarzeń takich jak zgon, nawrót choroby czy wyzdrowienie. Wyszukiwanie zależności między zmiennymi pozwalającymi określić czas wystąpienia zdarzenia można realizować przy użyciu podejścia sekwencyjnego pokrywania i reprezentacji w postaci reguł przeżycia [103]. Reprezentacja zbiorów danych dla celów analizy przeżycia z zastosowaniem systemów regułowych wymaga rozszerzenia tablicy decyzyjnej. Atrybutem decyzyjnym (zwanym też cenzurującym) jest binarna informacja, czy oczekiwane zdarzenie już zaszło, czy też jeszcze nie. Dodatkowo trzeba wyróżnić atrybut określający czas obserwacji. Pozostałe atrybuty warunkowe rozumie się tak, jak w przypadku zwykłej tablicy decyzyjnej.

W analizie przeżycia istotnym problemem jest obecność tzw. obserwacji uciętych lub cenzurowanych. Są to przykłady, co do których nie wiemy, czy zaszło dla nich interesujące nas zdarzenie. Przykładowo, analizując skuteczność terapii nowotworowej i jej wpływ na czas życia pacjentów, powiemy że wszystkie przykłady reprezentujące pacjentów którzy nadal żyją, są cenzurowane.

Przedstawioną w tej pracy metodę indukcji reguł akcji można zastosować także w zadaniu analizy przeżycia. Uzyskane reguły dostarczą informację o niezbędnych zmianach wartości atrybutów, które pozwolą na wydłużenie lub skrócenie czasu wystąpienia zdarzenia. Indukcja reguł akcji dla celów analizy przeżycia będzie przebiegać podobnie jak dla problemu regresji. Ogólny schemat indukcji reguł akcji opisany we wcześniejszych algorytmach należy zmienić by uwzględnił on fakt, że reguły przeżycia w swojej konkluzji zawierają funkcję estymującą krzywą przeżycia – estymatę Kaplana-Meira [50]. Sprawia to, że jeśli chcemy uzyskać reguły akcji, dające tylko lepsze krzywe przeżycia, do testów porównujących ze sobą konkluzję reguły źródłowej i docelowej należy zastosować odpowiedni test statystyczny, np. logarytmiczny test rang lub test Coxa.

Detale dotyczące indukcji przeżyciowych reguł akcji z zastosowaniem paradygmatu sekwencyjnego pokrywania opisano w pracy magisterskiej [45] napisanej pod kierunkiem dra Wróbla, promotora pomocniczego niniejszej pracy doktorskiej i nie będą one opisywane w dalszej części rozprawy.

---

**Algorytm 13** Specjalizacja reguły akcji regresji

---

**Wejście:**  $D(A, C)$  - zbiór przykładów (zbiór atrybutów  $A$  oraz ciągły atrybut decyzyjny  $C$ )

**Wejście:**  $D_s$  - zbiór niepokrytych jeszcze przykładów

**Wejście:**  $q$  - miara oceny jakości reguł klasyfikacyjnych

**Wejście:**  $mincov$  - minimalne dopuszczalne pokrycie reguły

**Wejście:**  $rel(r_1, r_2)$  - funkcja relacji konkluzji źródłowej i docelowej

**Wyjście:**  $r$  - reguła akcji regresji

**function** SPECJALIZUJREGUŁĘAKCJI( $D, D_s, q, mincov$ )

$r \leftarrow (\emptyset \rightarrow \emptyset)$

▷ Puste reguły regresji

$r_s \leftarrow (\emptyset \rightarrow \emptyset), \quad r_t \leftarrow (\emptyset \rightarrow \emptyset)$

**repeat**

$q_{best} \leftarrow -\text{inf}$

$w_{best_s} \leftarrow \text{NAJLEPSZYWARUNEKELEMENTARNY}(r_s, D, q, mincov)$

$r_s \leftarrow r_s \wedge w_{best_s}$

$atr \leftarrow \text{ATRYBUT}(w_{best_s})$

$W_t \leftarrow \text{WARUNKIELEMENTARNE}(\text{DLA}(\text{ATRYBUTU}(atr, D, mincov)))$

**for**  $w_t \in W_t$  **do**

$r_t \leftarrow r_t \wedge w_t$

**if**  $\text{KONKLUZJA}(r_s) \cap \text{KONKLUZJA}(r_t) \neq \emptyset$  **then**

continue

**end if**

**if**  $\neg rel(r_s, r_t)$  **then**

continue

**end if**

$q \leftarrow q(r_t, D)$

**if**  $q > q_{best}$  **then**

$q_{best} \leftarrow q, \quad w_{best_t} \leftarrow w_t$

**end if**

$r_t \leftarrow r_t \setminus w_t$

**end for**

$r_t \leftarrow r_t \wedge w_{best_t}$

▷ Rozszerz regułę akcją zbudowaną ze znalezionych warunków elementarnych

$r \leftarrow r \wedge \text{AKCJA}(w_{best_s}, w_{best_t})$

**until**  $(w_{best_s} = \emptyset)$

**return**  $r$

**end function**

---

## 4 Rekomendacje na podstawie zbiorów reguł akcji. Weryfikacja reguł akcji i rekomendacji.

Zbiory reguł akcji nie dają odpowiedzi na pytanie o to, jakie dokładnie działania należy podjąć względem pewnego obiektu, aby obiekt ten przeszedł z klasy źródłowej do docelowej. Reguły akcji wskazują jedynie jak muszą zmienić się wartości wybranych atrybutów, aby takie przejście było możliwe. W zastosowaniach praktycznych dobrze byłoby wiedzieć, jakie dokładnie działania w sensie fizycznym należy podjąć, aby osiągnąć efekt zmiany klasyfikacji obiektu - takie konkretne działania w literaturze nazywane są „meta-akcjami”. O tym podejściu do odkrywania „akcyjnych” wzorców w danych po raz pierwszy napisali Wang i in. [97], zaś termin meta-akcji został sformalizowany przez Rasia i Dardzińską w [73] i zastosowany do oceny skuteczności terapii chirurgicznych przez Touati i in. w [95].

Zakładając jednak, że informacja o koniecznej zmianie niesiona przez regułę akcji jest wystarczająca dla użytkownika do wpłynięcia na przykłady klasy źródłowej, otwartym problemem pozostaje wybór konkretnej reguły której przesłanka stanie się źródłem działań podjętych przez użytkownika. Zastosowanie paradygmatu sekwencyjnego pokrywania do indukowania reguł akcji pozwala nam uzyskać zwarte zbiory reguł, które jednak charakteryzują się tym, że może znaleźć się w nich więcej niż jedna reguła, której przesłanka pokryje analizowany przykład. Przez analogię do sposobu klasyfikacji regułami decyzyjnymi przy użyciu schematu głosowania lub najwyższego zaufania, jeśli chcemy wykorzystać reguły akcji do podjęcia rzeczywistych działań, należy najpierw rozwiązać ewentualne konflikty.

Innym interesującym zagadnieniem jest kolejność, w jakiej należy zmieniać wartości atrybutów (wpływać na obiekt tak, aby osiągnąć zmianę wartości atrybutu wskazaną w przesłance reguły akcji). Jest to ważne ze względu na fakt, że fizyczne wpływanie na obiekt celem zmiany wartości pewnego atrybutu może być kosztowne lub trudne.

Z tego względu warto wykonywać akcje poczynając od tej, która ma największy udział w przeniesieniu przykładu z klasy źródłowej do docelowej. Przez „udział akcji w przeniesieniu” należy rozumieć wsparcie w przykładach klasy źródłowej i docelowej danej akcji, ściśle wsparcie przesłanki akcji w przykładach źródłowych i wsparcie konkluzji akcji w przykładach docelowych. Reguły akcji bowiem mogą zawierać w sobie akcje dające większy i mniejszy udział w procesie przeniesienia przykładu, co obrazuje poniższy przykład.

**Przykład 4.0.1** *Rozważmy następującą akcję elementarną:  $A : V_{a_1} \rightarrow V_{a_2}$ , oraz dwa zbiory treningowe  $T1$  i  $T2$  zawierające po 50 przykładów klasy źródłowej  $s$  i 50 przykładów klasy docelowej  $t$ . W tabeli pokazano ile przykładów każdej z klas w danym zbiorze spełnia konkluzję i przesłankę akcji  $A$ .*

	$V_{a_1} \& s$	$V_{a_1} \& t$	$V_{a_2} \& s$	$V_{a_2} \& t$
$T1$	35	8	3	54
$T2$	90	1	3	6

*Jak widać ta sama akcja  $A$  w dwóch różnych zbiorach o tych samych atrybutach będzie wносить różny wkład do potencjalnego przeniesienia przykładu. W zbiorze  $T2$  wsparcie konkluzji akcji jest znikome, podczas gdy w zbiorze  $T1$  akcja  $A$  znajduje większe poparcie w przykładach.*

Idąc dalej, reguły akcji podają dość syntetyczny przepis na transformację obiektu. Dopóki użytkownik nie podejmie trudu wcielenia sugerowanych akcji w czyn, ponosząc przy tym pewien koszt, nie jesteśmy w stanie potwierdzić w żaden sposób, że sugerowane zmiany przyniosą oczekiwany rezultat. Jest to problem inherentny - reguły akcji podają jedynie predykcję zmiany, a ta może być zawodna. Dopóki nie zaobserwujemy oczekiwanego efektu, brak dowodów na skuteczność akcji. Można sobie wyobrazić sytuację, w której rzeczywiste obiekty są zmieniane zgodnie z sugestiami podawanymi przez reguły akcji wygenerowane na podstawie tych obiektów. Następnie śledzi się losy obiektu by stwierdzić, czy zmiany te przyniosły oczekiwany efekt (zmianę przyporządkowania obiekty do pożądanej klasy) czy też nie. Informacja o skuteczności wykonanych akcji, na podstawie obserwacji rzeczywistych obiektów posłużyłaby do oceny skuteczności algorytmu indukcji reguł akcji. Niestety, postępowanie takie może być nieopłacalne lub niemożliwe do wdrożenia. Powstaje więc potrzeba oszacowania zdolności reguł akcji do przenoszenia przykładów do pożądanej klasy bez konieczności zmieniania rzeczywistych obiektów, a na podstawie danych, które o tych obiektach posiadamy.

W dalszych częściach bieżącego rozdziału omówiono sposób uzyskiwania rekomendacji z gotowego zbioru reguł akcji oraz metodę pozwalającą estymować skuteczność zarówno reguł akcji jak i rekomendacji.

## 4.1 Zastosowanie reguł akcji i rekomendacji

Rekomendacje, podobnie jak zbiór reguł akcji, mogą zostać wykorzystane do zmiany właściwości obiektów. W tym celu należy „wykonać” przesłankę reguły na obiektach, których klasyfikację chcemy zmienić, tj. wykonać takie czynności, które sprawią że bieżące wartości atrybutów przykładowo zostaną zmienione na takie wartości, jakie sugeruje część docelowa przesłanki reguły akcji. O takim użyciu reguł akcji w dalszej części niniejszej pracy pisze się jako o zastosowaniu reguł akcji.

**Definicja 8** *Dana jest reguła akcji:*

$$r : IF(a_1, v_{s_1} \rightarrow v_{t_1}) \wedge \dots \wedge (a_m, v_{s_m} \rightarrow v_{t_m}) \textbf{ THEN } (d, s \rightarrow t) \quad (4.1)$$

oraz przykład

$$e : (a_1, v_{s_1}) \wedge \dots \wedge (a_m, v_{s_m}) \dots \wedge (a_n, v_{s_n}) \wedge (d, s). \quad (4.2)$$

*Poprzez zastosowanie reguły  $r$  na przykładzie  $e$  należy rozumieć wdrożenie takich działań, które sprawią że przykład  $e$  zostanie doprowadzony do postaci*

$$e' : (a_1, v_{t_1}) \wedge \dots \wedge (a_m, v_{t_m}) \dots \wedge (a_n, v_{s_n}) \wedge (d, ?). \quad (4.3)$$

Po zastosowaniu reguły  $r$  na przykładzie  $e$  nie jesteśmy w stanie nic powiedzieć o wartości atrybutu decyzyjnego  $d$ , nawet przy założeniu wysokiej precyzji docelowej części reguły, która została zastosowana. Zastosowanie reguły jest realizacją działań wynikających z obserwacji dostępnych w danym zbiorze danych, które opisują pewną populację czy zjawisko. Nie mamy gwarancji, że przynależność przykładów do pewnej klasy decyzyjnej jest w pełni zdeterminowana przez dostępne atrybuty. Być może decydujący czynnik jest nieznan, lub jest nim kombinacja większej niż dostępna do analizy liczby cech. Wówczas może dojść do sytuacji, że mimo iż przypisanie przykładów do klasy w zbiorze treningowym jest poprawne, to reguła akcji nie będzie w stanie wskazać transformacji gwarantującej przeniesienie przykładu do pożądanej klasy. Innymi słowy, proces sugerowania akcji przez opisane metody jest zawodny. Faktyczny skutek zastosowania reguł na pewnych przykładach może zostać

poznany po wdrożeniu fizycznych działań, które doprowadzą parametry rzeczywistych obiektów do wartości wskazywanych przez reguły akcji lub rekomendacje. Działanie takie może być obciążone wysokim kosztem lub czas oczekiwania na wyniki zmian może być bardzo długi. Tym samym określenie, czy obiekt, który został poddany sugerowanym zmianom został w wyniku przeniesiony do pożądanej klasy, to jest - czy sugerowane zmiany przynoszą oczekiwane działanie, może odbyć się dopiero po ponownym zbadaniu (np. przez eksperta dziedzinowego) przetransformowanego obiektu.

Podobny problem spotyka się przy analizie wyników działania klasyfikatorów. Jeśli nie znamy rzeczywistej klasy badanego przykładu, dostajemy tylko wynik działania klasyfikatora. Klasyfikator wnioskuje opierając się o dane pochodzące ze zbioru przykładów treningowych. Dane takie mogą być niekompletne, i subtelne związki między wartościami parametrów rzeczywistych obiektów a ich przynależnością do pewnej klasy mogą zostać nieodkryte przez klasyfikator. W takiej sytuacji, aby określić czy odpowiedź klasyfikatora była poprawna, należy zasięgnąć opinii ekspertów lub wyników działania innej, niezawodnej metody, określania klasy badanych obiektów - potencjalnie oba te działania niosą za sobą pewien koszt, który chcemy zminimalizować właśnie przez stworzenie klasyfikatora, który będąc modelem komputerowym może w sposób szybki i tani pomóc określić klasę pewnych obiektów na podstawie ich opisów.

W ujęciu problemu klasyfikacji klasycznym wyjściem z tej problematycznej sytuacji jest określenie poziomu zaufania do wyników klasyfikatora poprzez sprawdzenie jego dokładności, precyzji lub innych pożądanych parametrów na zbiorze testowym. Zbiór testowy najczęściej tworzą przykłady wydzielone w odpowiedni sposób ze zbioru treningowego, dane historyczne, dla których istnieje pewne przypisanie każdego z przykładów do właściwej klasy decyzyjnej. Innymi słowy, zbiór testowy i treningowy. Znając wybrane parametry jakości klasyfikatora, jego użytkownik wie z jakim ryzykiem pomyłki musi się liczyć, używając tego klasyfikatora zamiast innej, niezawodnej metody. Zbliżone podejście można zastosować w kontekście reguł akcji i rekomendacji, o czym traktuje dalsza część tego rozdziału. Podejście to jest inspirowane procesem walidacji krzyżowej (sprawdzianu krzyżowego) stosowanego do oceny jakości klasyfikatorów. Zasadnicza różnica polega na tym, że w problemie klasyfikacji oceniamy klasyfikator znając rzeczywiste przyporządkowanie przykładów do klas. Możemy wtedy łatwo stwierdzić, jak często klasyfikator popełnia błąd i na tej podstawie wyznaczyć poziom ufności do jego decyzji. W przypadku zastosowania reguł akcji



lub rekomendacji otrzymujemy nowy przykład. Przykład taki, będący wynikiem zastosowania akcji na przykładzie oryginalnym, nazywać będziemy przykładem zmutowanym lub przetransformowanym. Potencjalnie ten nowy, zmutowany przykład jest niepowtarzalny - nie znajdziemy dla niego odpowiednika o takim samym zestawie wartości atrybutów w zbiorze treningowym. Nie dysponujemy statycznym źródłem danych, które umożliwiłoby określenie skuteczności mutacji przykładów poprzez porównanie ich ze wzorcem, jak to się dzieje w procesie oceny klasyfikatorów. W podrozdziale 4.3 podano sposób szacowania efektywności mutacji przykładów wykorzystujący klasyfikator jako arbitra decydującego o tym, czy mutacja pojedynczego przykładu była skuteczna, czy też nie.

Niezależnie, czy jako źródło akcji do wykonania przyjmiemy zbiór reguł akcji czy metodę podającą rekomendacje, należy określić sposób decydowania o tym, które akcje zostaną wykonane.

Jeżeli zachodzi potrzeba wdrożenia w życie sugestii wskazywanych bezpośrednio przez zbiór reguł akcji, należy podjąć następujące kroki:

1. Dla wybranego przykładu  $e$  znaleźć wszystkie reguły akcji, których źródłowa część pokrywa ten przykład;
2. Spośród wyselekcjonowanych w kroku 1. reguł wybrać tę, która zostanie zastosowana, poprzez np. wybranie reguły o największym pokryciu, największej precyzji części źródłowej, najlepszej dokładności części docelowej lub zgodnie z innym kryterium właściwym do rozwiązywanego problemu;
3. Zastosować wybraną regułę na przykładzie  $e$ .

To postępowanie ma na celu rozwiązanie potencjalnych konfliktów między regułami. Jest to podejście podobne do schematu największego zaufania opisanego w podrozdziale 2.2.2.

Wykorzystując podany w dalszej części tego rozdziału sposób uzyskiwania rekomendacji, postępowanie to zmienia się:

1. Z istniejącego zbioru reguł akcji należy wygenerować meta-tablicę  $mE$ ;
2. Dla wybranego przykładu  $e$  przeprowadzić proces wspinaczki po  $mE$  w celu uzyskania rekomendacji;
3. Zastosować uzyskaną rekomendację na przykładzie  $e$

Oczywiście, w przypadku indukcji wielu rekomendacji dla jednego przykładu, należy jeszcze wybrać ostateczną rekomendację, która zostanie zastosowana, co w zasadzie ponownie wprowadza problem konfliktu. Indukcja wielu rekomendacji dla jednego przykładu jest jednak krokiem opcjonalnym w podanym dalej algorytmie.

## 4.2 Meta-reguły

Akcje elementarne budujące reguły akcji w wyindukowanym zbiorze mogą być niedokładne, tj. mogą pokrywać jednocześnie przykłady klasy źródłowej i docelowej, ale mimo tej niedoskonałości niosą ze sobą informację o tym, w jakich zakresach wartości danego atrybutu skupiają się reprezentanci poszczególnych klas. Dalsza eksploracja tej informacji powinna pozwolić ustalić zakresy jak najbardziej zbliżone do zakresów dokładnych. Tworząc reguły akcji tylko w oparciu o te (potencjalnie) dokładne zakresy oraz ważąc akcje zgodnie z regułą, że im bardziej dokładne zakresy, tym bardziej dana akcja jest preferowana, jesteśmy w stanie wygenerować rekomendację odnośnie kolejności i sposobu, w jaki należy wpływać na atrybuty obiektów w celu osiągnięcia pożądanego efektu zmiany zakwalifikowania przykładu do klasy.

Uzyskane w toku takiego działania rekomendacje mają postać reguł akcji. Aby podkreślić, że powstały one na podstawie analizy innych reguł, nazywane będą zamiennie meta-regułami. Meta-reguły mogą być tworzone zarówno na podstawie zbiorów reguł akcji, jak i dwóch zbiorów reguł klasyfikacyjnych, pod warunkiem że zbiory te zawierają reguły wskazujące na rozłączne klasy. Opisane tutaj rekomendacje lub meta-reguły akcji są równoważne rekomendacjom rozważanym w [96], można o nich myśleć jak o specjalizowanych regułach akcji tworzonych dla pojedynczego przykładu, mających na celu wskazanie najlepszych akcji prowadzących do zmiany jego przyporządkowania do klasy docelowej.

W dalszej części tego rozdziału opisano sposób uzyskiwania meta-reguł akcji, który składa się z dwóch procesów: budowy meta-tablicy oraz procesu przeszukiwania w tej tablicy celem wygenerowania akcji elementarnych, które będą budować przesłankę rekomendacji. Opisana metoda pozwala uzyskać dla danego przykładu więcej niż jedną rekomendację, a kolejne akcje elementarne dodawane są do przesłanki rekomendacji zgodnie z kryterium ich istotności, tzn. pierwsza w kolejności akcja będzie akcją o największym potencjale do przeniesienia przykładu do klasy docelowej (jak w przykładzie 4.0.1).

### 4.2.1 Meta-tablica

Zdefiniowanie rekomendacji rozpoczyna się od budowy meta-tablicy. Meta-tablica ma na celu wyznaczenie, w ramach każdego atrybutu warunkowego, takich zakresów (przedziałów) wartości, które w dalszym toku pozwolą na utworzenie jak najbardziej dokładnych akcji. Do jej utworzenia wykorzystane mogą być wcześniej wyindukowane reguły akcji lub reguły decyzyjne wskazujące na klasę źródłową i docelową. Wszystkie warunki elementarne, tj. suma zbiorów  $W_{R_S} \cup W_{R_T}$  grupowane są zgodnie z atrybutem decyzyjnym na którym są oparte. Następnie, w ramach każdej z grup, wyszukuje się wszystkie możliwe przecięcia między zakresami tworzącymi warunki elementarne na danym atrybucie, efektywnie tworząc nowe, węższe zakresy. Nadto dodaje się zakresy brzegowe, które zaczynają (kończą) się wartością dolną (górną) zbioru wartości danego atrybutu.

**Przykład 4.2.1** *Atrybut temperatura, przyjmuje wartości ze zbioru liczb rzeczywistych, oraz dane są trzy warunki elementarne zbudowane na tym atrybucie:*

*temperatura  $\leq -10.0$ , temperatura  $\in (-15.0, 25.0]$  oraz temperatura  $> 15.0$ .*

*Po wyszukaniu wszystkich przecięć i dodaniu zakresów brzegowych, uzyskuje się następującą listę przedziałów:  $(-\infty, -15.0)$ ,  $(-15.0, -10]$ ,  $(-10.0, 15.0]$ ,  $(15.0, 25.0]$ ,  $(25.0, \infty)$ . Ponieważ atrybut decyzyjny jest znany, zakresy te łatwo można przekształcić do postaci warunków elementarnych, np. (temperatura  $\leq -15.0$ ).*

*W przypadku atrybutów nominalnych nie ma potrzeby wyznaczania przecięć, a nowy zbiór zakresów jest tożsamy z wszystkimi dopuszczalnymi wartościami danego atrybutu.*

Wraz z każdym uzyskanym zakresem skojarzona jest liczba przykładów klasy źródłowej oraz docelowej, którą dany zakres pokrywa. Dodatkowo z każdym zakresem skojarzone są reguły, w których przesłankach występuje ten zakres (warunki elementarne budujące przesłankę pokrywają dany zakres). W zastosowaniu praktycznym utrzymuje się dwie listy reguł, jedną zawierającą reguły wskazujące na klasę źródłową i drugą zawierającą reguły wskazujące na klasę docelową.

Po wyznaczeniu nowych zakresów i przekształceniu ich do postaci warunków elementarnych w ramach każdego z dostępnych atrybutów uzyskujemy tyle zbiorów warunków elementarnych, ile było dostępnych atrybutów. W następnym kroku wyznacza się iloczyn kartezjański wszystkich tych zbiorów i zapisuje się go w postaci tablicy, nazywanej meta-tablicą. Każdy z wierszy w meta-tablicy nazywany będzie

meta-przykładem. Każdy meta-przykład zawiera po jednym warunku elementarnym dla każdego atrybutu warunkowego w analizowanym zbiorze. Kolumny meta-tablicy nazywać będziemy meta-atrybutami. W meta-tablicy brak jest atrybutu decyzyjnego.

**Przykład 4.2.2** *Przyjmijmy, że do zbudowania meta-tablicy dysponujemy następującymi warunkami elementarnymi atrybutu temperatura:  $(temperatura \leq -15.0)$ ,  $(temperatura \in (-15.0, 0.0])$ ,  $(temperatura \geq 15.0)$  oraz dwoma warunkami elementarnymi atrybutu wilgotność:  $(wilgotność, wysoka)$  i  $(wilgotność, niska)$ .*

*Meta-tablica przyjmie następującą formę:*

<i>id</i>	<i>temperatura</i>	<i>wilgotność</i>
1	$(temperatura \leq -15.0)$	$(wilgotność, wysoka)$
2	$(temperatura \leq -15.0)$	$(wilgotność, niska)$
3	$(temperatura \in (-15.0, 0.0])$	$(wilgotność, wysoka)$
4	$(temperatura \in (-15.0, 0.0])$	$(wilgotność, niska)$
5	$(temperatura \geq 15.0)$	$(wilgotność, wysoka)$
6	$(temperatura \geq 15.0)$	$(wilgotność, niska)$

Meta-tablica zawiera wszystkie możliwe kombinacje wszystkich zakresów jakie wystąpiły w regułach będących danymi wejściowymi algorytmu, z tym że zakresy te zostały ograniczone tak, aby przecięcie dwóch dowolnie obranych zakresów w ramach pewnego atrybutu było zawsze puste.

Formalnie konstrukcję meta-tablicy można zdefiniować następująco: dane są zbiór reguł akcji  $R$  i zbiór przykładów  $E(A, d)$ . Meta-tablica  $mE = (mA, \{\})$  składa się ze zbiorów meta-atrybutów  $mA = \{{}_m a_{i_1}, {}_m a_{i_2}, \dots, {}_m a_{i_m}\}$ . Jeśli atrybut  $a \in A$  nie jest obecny w przesłance żadnej z reguł w  $R$ , to nie jest też obecny jego odpowiednik w zbiorze meta-atrybutów  $mA$ .

Zbiór  $mA$  jest złożony z meta-atrybutów, których konstrukcja zależna jest od typu atrybutu  $a$ , z którego tworzony jest dany meta-atrybut.

Jeśli pewien atrybut symboliczny  $a \in A$  występuje w przynajmniej jednej akcji elementarnej w  $R$ , to atrybut ten przechodzi do zbioru  $mA$  w niezmienionej formie, tj.  ${}_m a := a$ .

Inaczej postępuje się w przypadku atrybutów numerycznych. Każda z reguł w  $R$  może być przedstawiona w formie  $r_s \rightarrow r_t$ . Niech  $a \in A$  oznacza atrybut numeryczny, który występuje w przynajmniej jednej akcji elementarnej wśród reguł z  $R$ . Aby uzyskać z niego meta-atrybut  ${}_m a$  stosuje się następujące kroki: Wyznacza się zbiór  $W_{R_{a_S}}$  ( $W_{R_{a_T}}$ ) zawierający wszystkie warunki elementarne zbudowane na atrybucie  $a$

z przesłanek wszystkich reguł  $r_s$  ( $r_t$ ) w zbiorze  $R$ . W dalszym kroku wyznacza się zbiór  $W_a = W_{R_{a_S}} \cup W_{R_{a_T}}$ , który zawierają wszystkie warunki elementarne zbudowane na atrybucie  $a$ , które wystąpiły w regułach w  $R$ . Zbiory  $W_{R_{a_S}}$  i  $W_{R_{a_T}}$  mogą być wyznaczone na podstawie zbiorów  $W_{R_S}$  i  $W_{R_T}$  zdefiniowanych jak w 6 poprzez wybranie z nich warunków elementarnych zbudowanych na atrybucie  $a$ .

Warunki elementarne budujące  $W_a$  można przedstawić w formie  $a > v$  lub  $a \leq v$ , gdzie  $v$  jest pewną wartością z dziedziny atrybutu  $a$ . Sortując w porządku rosnącym wartości wszystkie wartości  $v$  występujące w  $W_a$  uzyskujemy podział wartości atrybutu  $a$ . Każdy z zakresów  $(v_i, v_{i+1}]$ , gdzie  $v_i < v_{i+1}$ , otrzymuje kolejny numer porządkowy. Ten numer porządkowy reprezentuje wartość meta-atrybutu  ${}_m a$  dla danego meta-przykładu.

Konstruując meta-atrybuty zgodnie z podanym wyżej opisem, oraz przyjmując że  ${}_m V_{a_i}$  oznaczać będzie zbiór wartości meta-atrybutu, możemy określić meta-tablicę jako iloczyn kartezyjański zbiorów  ${}_m V_{a_1} \times {}_m V_{a_2} \times \dots \times {}_m V_{a_n}$ . Każdy meta-przykład jest m-wymiarową krotką w tak zdefiniowanej przestrzeni.

## 4.2.2 Indukcja rekomendacji

Rekomendacja przyjmuje formę reguły akcji. Jest ona jednak konstruowana pod kątem konkretnego przykładu. W tej sekcji opisano sposób konstrukcji rekomendacji przy zastosowaniu strategii wspinaczki. Przeszukiwaną strukturą danych będzie meta-tablica opisana w poprzedniej sekcji.

Aby wyznaczyć rekomendację dla pewnego przykładu  $e$  pochodzącego ze zbioru  $E(A, d)$  potrzebny jest zbiór reguł wyindukowany wcześniej na zbiorze treningowym właściwym dla takiego celu, tj. pochodzącym z podziału zbioru  $E$ , lub innego zbioru opisanego tymi samymi atrybutami i reprezentującego próbkę pochodzącą z tej samej populacji. Niezbędne jest także zachowanie zbioru przykładów, na którym zostały wyindukowane reguły akcji użyte do stworzenia meta-tablicy. Zbiór ten będzie niezbędny w trakcie wspinaczki do oceny jakości indukowanej rekomendacji.

Wynikowa rekomendacja będzie mieć postać reguły akcji. Część źródłowa tej reguły zostanie zbudowana z wpisów znajdujących się w meta-tablicy, które pokrywają wartości poszczególnych atrybutów budujących przykład.

**Przykład 4.2.3** *Przyjmijmy, że dysponujemy meta-tablicą taką samą, jak opisaną w przykładzie 4.2.2, oraz przykładem  $(temperatura, -17.5) \wedge (wilgotność, niska)$ . Aby wyznaczyć źródłową część wynikowej rekomendacji, w meta-tablicy należy odszukać*

*meta-przykład, który pokrywa zaprezentowany przykład. W tym przypadku będzie to*  $(temperatura \leq -15.0) \wedge (wilgotność, niska)$ .

Warto zwrócić uwagę, że zgodnie z przedstawioną wcześniej definicją meta-tablicy, jeśli w pewnym zbiorze występują tylko atrybuty nominalne, to źródłowa część rekomendacji jest tożsama z analizowanym przykładem. W przypadku atrybutów numerycznych, źródłowa część rekomendacji będzie się składać z warunków elementarnych prezentujących zakresy poszczególnych atrybutów pokrywające analizowany przykład.

Docelowa część rekomendacji jest wyznaczana w procesie wspinaczki sterowanej wybraną przez użytkownika miarą oceny jakości reguł klasyfikacyjnych. Wartości w poszczególnych kolumnach meta-tablicy mogą być reprezentowane jako warunki elementarne, wobec czego ich złożenie (koniunkcja) może być traktowana jak przesłanka reguły klasyfikacyjnej.

Pierwszym krokiem indukcji rekomendacji jest wyznaczenie źródłowej części wynikowej rekomendacji, zgodnie z podanym wyżej opisem. Proces ten w żaden sposób nie wpływa na część docelową. Wyznaczanie części docelowej rozbite jest na dwa pod-procesy – wzrost i przycinanie.

W fazie wzrostu, meta-tablica przeglądana jest w poszukiwaniu wartości (warunku elementarnego), który zagwarantuje możliwie największy wzrost wartości miary oceny jakości. Jakość wyznaczana jest poprzez zmierzenie pokrycia wybranego warunku elementarnego na zbiorze przykładów, które posłużyły do stworzenia reguł akcji z których powstała meta-tablica. Pomiaru tego dokonuje się tak, jakby klasą pozytywną była klasa docelowa. W wyniku uzyskuje się wartości  $p, n, P, N$ , które służą do wyliczenia wartości funkcji oceny jakości. Po wyznaczeniu pierwszego, najlepszego warunku elementarnego, eliminuje się kolumnę (meta-atrybut) z której on pochodzi z dalszych poszukiwań. W dalszym ciągu szuka się nowego warunku elementarnego, który po dołączeniu do już znalezionego warunku spowoduje dalszy wzrost jakości. Wybrane warunki elementarne tworzą tymczasową przesłankę. Proces ten jest kontynuowany tak długo, jak możliwy jest wzrost jakości poprzez dołączanie kolejnych warunków elementarnych.

Po zakończeniu fazy wzrostu następuje przycinanie. W trakcie tego procesu kolejno wyłącza się warunki elementarne przesłanki zbudowanej w fazie wzrostu. Jeśli zignorowanie danego warunku elementarnego nie ma wpływu na wartość funkcji jakości, lub powoduje jej wzrost - warunek taki jest usuwany z tymczasowej przesłanki na stałe. Postępowanie to trwa dopóki istnieją takie warunki, które można usunąć z przesłanki

z pozytywnym skutkiem (tj. wzrostem lub utrzymaniem wartości) dla jakości całej przesłanki. Uzasadnienie tego postępowania jest analogiczne do przycinania reguł klasyfikacyjnych.

Ostatecznie uzyskujemy przesłankę - koniunkcję warunków elementarnych - która charakteryzuje się wysoką skutecznością wskazywania przykładów klasy docelowej w zbiorze treningowym, zgodnie z zadaną przez użytkownika miarą, która powinna być dostosowana do celów użytkownika metody i specyfiki analizowanego problemu.

Ostatecznie użytkownikowi prezentowana jest rekomendacja w postaci reguły akcji, która powstaje poprzez złączenie części źródłowej i docelowej wyznaczonych zgodnie z powyższym opisem. Rekomendacja jest formowana przez utworzenie akcji elementarnych z warunków elementarnych budujących część źródłową i docelową rekomendacji opartych o ten sam atrybut. Rekomendacja taka, ze względu na fakt powstanie w wyniku analizy meta-tablicy może być także nazywana meta-regułą akcji.

Opisaną metodę można postrzegać jako modyfikację algorytmu indukcji reguł akcji. Część źródłowa rekomendacji jest wyznaczana wprost na podstawie przykładu, dla którego generowana jest rekomendacja. Stanowi ją koniunkcja meta-wartości pokrywających dany przykład. Koniunkcję tę można potraktować jako przesłankę reguły źródłowej. Teraz, znając pełną przesłankę reguły źródłowej, przeglądamy ją od lewej do prawej, i dla każdego warunku elementarnego wyszukujemy w meta-tablicy najlepszego warunku elementarnego, tworząc coraz to dłuższą przesłankę reguły docelowej. Wybór warunków jest sterowany miarą jakości i poprzez ocenę budowanej przesłanki na zbiorze treningowym. Porównując tę modyfikację z algorytmem oryginalnym, można powiedzieć że metoda indukcji reguł akcji dynamicznie dołącza kolejne akcje elementarne do przesłanki reguły akcji, a metoda indukcji rekomendacji dla znanej już przesłanki reguły źródłowej dobiera najlepszą możliwą regułę docelową, używając meta-tablicy jako struktury danych. Należy podkreślić, że meta-tablica zawiera już zdyskretyzowane wartości atrybutów ciągłych, zaś podczas rozszerzenia przesłanek reguł dyskretyzacja atrybutów ciągłych jest wykonywana na bieżąco.

Przedstawiona metoda uzyskiwania rekomendacji dokonuje tylko jednokrotnego przejścia meta-tablicy w poszukiwaniu najlepszych docelowych warunków elementarnych. Można zaproponować prostą modyfikację algorytmu, która umożliwi uzyskiwanie wielu rekomendacji dla jednego przykładu. Polega ona na tym, że po zakończeniu fazy wzrostu i przycinania pierwszej rekomendacji z meta-tablicy usuwa się wszystkie wystąpienia warunków elementarnych, które znajdują się w docelowej

części tej rekomendacji. Tak zubożona meta-tablica jest ponownie podawana na wejście algorytmu wspinaczki wraz z analizowanym przykładem. Proces ten jest kontynuowany tak długo, dopóki możliwe jest jeszcze utworzenie kolejnych rekomendacji. Ewentualnie algorytm można sparametryzować tak, aby zwracał maksymalną zdefiniowaną przez użytkownika liczbę rekomendacji.

Uzyskiwanie wielu rekomendacji niejako przywraca problem konieczności rozwiązania konfliktów podczas wybierania najlepiej pasującej reguły. Z drugiej strony, każda z wytworzonych meta-reguł jest specjalizowana pod konkretny przykład. W zależności od potrzeb użytkownika, może okazać się że rekomendacje wygenerowane w kolejnych iteracjach charakteryzują się mniejszym kosztem, zawierają mniej akcji lub są bardziej zwarte i łatwiejsze w zrozumieniu, a przy tym nadal składają się z możliwie dokładnych warunków elementarnych.

Metoda uzyskiwania rekomendacji jest otwarta na dalsze kryteria zawężające przeszukiwania, którym jednak nie poświęcimy wiele miejsca w tej pracy. Jedną z proponowanych modyfikacji jest indukowanie tylko takich rekomendacji, które składają się z akcji sugerujących „niewielkie” zmiany w wartościach atrybutów - w przypadku atrybutów numerycznych mamy na myśli minimalizowanie odległości euklidesowej między zakresem źródłowym a docelowym akcji, w przypadku atrybutów symbolicznych konieczne byłoby zdefiniowanie macierzy kosztów przejścia między wartościami atrybutów. Inną propozycją jest indukowanie rekomendacji o ograniczonej liczbie akcji, co może pozytywnie wpłynąć na czytelność uzyskanych rekomendacji, szczególnie w przypadku analizy zbiorów składających się z wielu atrybutów.

Aby uzyskać rekomendacje dla zadania regresji, należy zmodyfikować proces wzrostu i przycinania reguły. Ponieważ używana jest ta sama struktura danych, w przeszukiwaniu jej wystarczy zmienić sposób wyboru najlepszego warunku elementarnego, który ma rozszerzyć (zostać usunięty w przypadku przycinania) budowaną rekomendację. W zadaniu regresji nie mamy klas, wobec czego nie można zmierzyć pokrycia budowanej rekomendacji w odniesieniu do klasy docelowej. Zamiast tego stosuje się miarę jakości wyznaczoną na pokryciu reguły wyznaczonym jak dla standardowej reguły regresji. Nakładany jest dodatkowy warunek przy wyborze rozszerzającego warunku elementarnego: użytkownik metody może zadać, że interesujące są tylko rekomendacje, które zapewnią wyższą lub niższą wartość atrybutu decyzyjnego. Domyślnym warunkiem jest zapewnienie, że wartość sugerowana przez rekomendację jest różna od aktualnej wartości atrybutu decyzyjnego w analizowanym



przykładzie. Jeśli kandydujący rozszerzający warunek elementarny po dołączeniu do przesłanki reguły i przeliczeniu jej pokrycia powoduje naruszenie zadanego warunku, jest odrzucany, nawet jeśli zapewniałby wzrost jakości rekomendacji. Sama wartość atrybutu decyzyjnego wskazywanego przez rekomendację jest ustalana w następujący sposób: wszystkie wybrane meta-wartości są transformowane do postaci przesłanki klasycznej reguły regresyjnej, dla której wyznacza się pokrycie w zbiorze treningowym. Na tej podstawie ustalana jest wartość atrybutu decyzyjnego oraz  $\epsilon_1, \epsilon_2$ , które to budują konkluzję rekomendacji.

### 4.3 Weryfikacja reguł akcji i rekomendacji

Celem indukcji reguł akcji może być poznanie związków między zmianami wartości atrybutów a zmianą przypisania przykładów do klasy decyzyjnej (funkcja opisowa) lub uzyskanie wiedzy o możliwych działaniach, które przyniosą zakładaną korzyść (tzw. *Actionable Knowledge*). Drugi wspomniany cel przez niektórych traktowany jest także do określenia jak bardzo odkryty wzorzec jest interesujący. W [86] zdefiniowano dwa subiektywne wskaźniki, pozwalające określić jak bardzo pewien wzorzec jest interesujący: jego nieoczekiwaność (ang. *unexpectedness*) i wykonalność (ang. *actionability*). Przez „wykonalność” rozumiemy, że pewien wzorzec reprezentujący wiedzę jest interesujący dla użytkownika, ponieważ znając go, użytkownik może zareagować i odnieść dzięki temu korzyść. Możliwość podjęcia działań przynoszących korzyść organizacji lub osobom została nawet uznana za ostateczny cel przeprowadzania eksploracji danych przez autorów pracy przeglądowej [43]. Niewątpliwie możliwość odkrywania działań prowadzących do poprawy pewnej sytuacji wprost z opisu bieżącego stanu obiektów ma potencjał do zastosowań praktycznych.

Przedstawione w niniejszej pracy metody są zawodne, bo - jak wszystkie modele AI/ML - są obciążone indukcyjnie. Podobnie jak metody klasyfikacji i regresji, tak też i reguły akcji nie zawsze dają pewną odpowiedź, którą można wcielić w życie z nadzieją na osiągnięcie pożądanego wyniku. Zbiory reguł akcji i silnik rekomendacji tworzą model wiedzy o możliwych do podjęcia akcjach oraz spodziewanych skutkach tych akcji. W dalszej części model taki będzie nazywany „modelem akcji” lub „modelem akcyjnym”, gdy nie ma znaczenia czy mowa o zbiorach reguł akcji lub rekomendacjach. Badając odpowiedź takiego modelu na znanych wcześniej danych jesteśmy w stanie określić zdolność modelu do sugerowania skutecznych akcji,

podobnie jak badając jakość klasyfikatora na zbiorze walidacyjnym określamy jego przydatność do zastosowania w zadaniu klasyfikacji.

W przypadku zagadnienia klasyfikacji, zazwyczaj przyjmuje się następującą metodę. Ze zbioru przykładów, w którym znamy przypisanie do klas, wydziela się zbiór treningowy i testowy. Przykładów ze zbioru treningowego używa się do wytrenowania (tj. ustalenia wewnętrznych parametrów) klasyfikatora, zaś na zbiorze testowym przeprowadza się próbną klasyfikację obiektów. Na tej podstawie można wyznaczyć macierz pomyłek klasyfikatora i określić jego jakość. Przykłady ze zbioru testowego nie były podane na wejście klasyfikatora w trakcie treningu, wobec czego są one traktowane jak przykłady nieznanne, dobrze odzwierciedlając faktyczny cel trenowania klasyfikatora. Czasami wprowadza się także trzeci zbiór, zwany walidacyjnym, który służy do dodatkowego strojenia parametrów klasyfikatora, jeszcze przed sprawdzeniem go na zbiorze testowym. Działanie takie ma pomóc uniknąć zjawiska nadmiernego dopasowania się klasyfikatora (ang. *overfitting*) do danych treningowych. Stosowane jest także podejście k-krotnej walidacji krzyżowej, gdzie zbiór przykładów dzieli się na k równolicznych podzbiorów, następnie kolejno każdy z k podzbiorów traktuje się jako zbiór testowy, a wszystkie pozostałe razem jako zbiór treningowy. Uzyskuje się wówczas k zestawów wyników, które należy uśrednić lub połączyć w inny sposób by móc ocenić klasyfikator. Pochodną podejścia k-krotnej walidacji krzyżowej jest *leave-one-out*, gdzie wartość k ustala się równą liczbie przykładów w zbiorze - podejście to stosuje się w szczególnych przypadkach.

Dzieląc przykłady na zbiory, niezależnie od wybranej metody, należy pamiętać o ich stratyfikacji, tzn. aby proporcja przykładów z różnych klas została zachowana w utworzonych podzbiórach. Jest to szczególnie ważne przy testowaniu klasyfikatora na danych niezbalansowanych, tj. takich, w których pewna klasa występuje częściej niż inne [8]. Jeśli odpowiednie proporcje częstości występowania przykładów nie zostaną przeniesione na zbiory treningowy, walidacyjny i testowy lub zbiory występujące w sprawdzaniu krzyżowym, uzyskanie dobrego klasyfikatora może być niemożliwe, gdyż np. będzie trenowany na zbiorze „łatwiejszym” niż oryginalny.

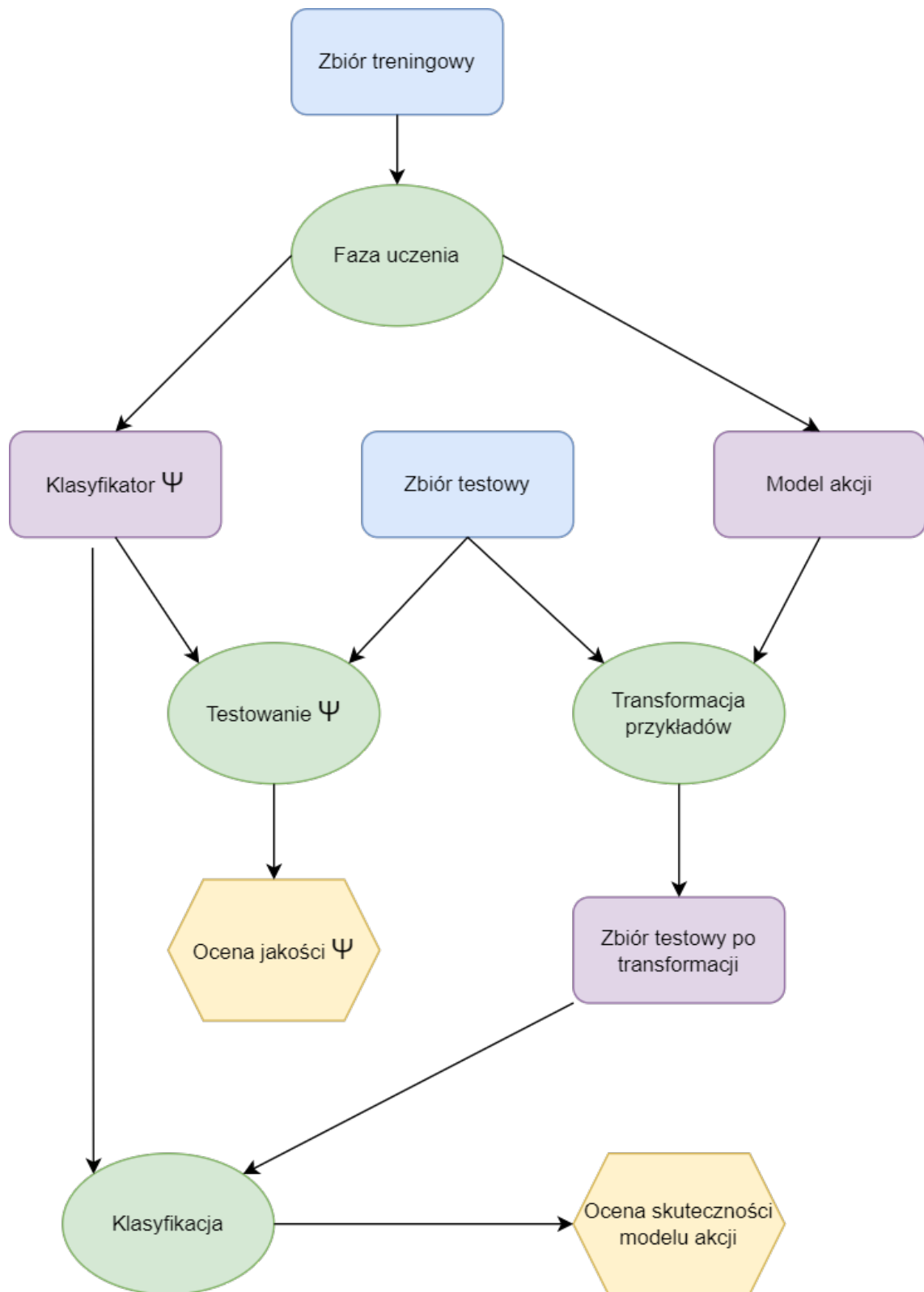
Koncepcje znane z zagadnienia oceny jakości klasyfikatorów były inspiracją do zaproponowania sposobu oceny modeli akcyjnych, tj. zbiorów reguł akcji i rekomendacji. Celem oceny takiego modelu jest uzyskanie informacji na temat tego jak pewne są akcje sugerowane przez badany model, lub inaczej: czy uzyskanie oczekiwanego skutku po wykonaniu akcji sugerowanych przez model znajduje pokrycie w danych?

W przeciwieństwie do modeli klasyfikacyjnych, wynik działania modelu akcyjnego nie będzie przedstawiany w postaci macierzy pomyłek i wyznaczonych na jej podstawie miar oceny jakości klasyfikacji. Reguła akcji swoją przesłanką wydziela jednocześnie przykłady klasy źródłowej i docelowej, w związku z czym trudno scharakteryzować jej jakość jedną wartością liczbową. Podobnie cały zbiór reguł akcji lub silnik rekomendacji w odpowiedzi na podany do analizy przykład zwraca najlepiej pasującą regułę akcji lub utworzoną rekomendację, co nie pozwala na bezpośrednie zaliczenie takiej odpowiedzi jako „dobra” lub „zła”, co ma miejsce w przypadku oceny modeli klasyfikacyjnych.

Przedstawiony dalej schemat eksperymentu weryfikującego pozwala na uzyskanie kilku wskaźników numerycznych określających jakość badanego modelu, poprzez symulowane wykonanie akcji na przykładach ze zbioru testowego i sprawdzenie w jaki sposób zostaną zaklasyfikowane tak zmodyfikowane przykłady. Postępowanie to ma na celu ustalenie punktu odniesienia pozwalającego ustalić, czy sugerowane akcje są skuteczne. Zakładamy, że nie mamy informacji historycznych o tym, jakie akcje były wykonywane na obiektach i jaki przyniosły rezultat, ani nie jesteśmy w stanie śledzić losów obiektu po wdrożeniu zasugerowanych zmian. Innymi słowy nie dysponujemy źródłem prawdy na temat skutków modyfikacji badanych obiektów. Do celów oceny modelu akcyjnego, takie źródło zostanie utworzone sztucznie. Jego rolę będzie pełnić klasyfikator, który przyjmie rolę arbitra stwierdzającego, do jakiej klasy przynależy przykład, który został zmodyfikowany przez akcje podane przez model akcyjny. W ten sposób będziemy w stanie określić, dla których przykładów model podał skuteczne akcje, a dla których nie, co pozwoli nam liczbowo określić jakość całego modelu akcyjnego. Ze względu na ważkość skutecznej klasyfikacji w tym procesie, należy użyć metody o najlepszej możliwej skuteczności klasyfikacji dla danego problemu. Klasyfikator taki można wytrenować na tym samym zbiorze, na podstawie którego utworzono model akcyjny, lecz nie jest to konieczne, o czym krótko wspomniano w dalszej części pracy.

Schemat postępowania w celu uzyskania wskaźników skuteczności badanych modeli akcji przedstawiono na rysunku 4.1, na którym owalami oznaczono niezbędne do wykonania przez użytkownika procesy, prostokąty oznaczają dane: wejściowe (kolor niebieski), oraz pośrednie (kolor fioletowy), a wynikowe statystyki skuteczności modelu akcji oraz jakości użytego klasyfikatora reprezentowane są sześciokątami.

Faza uczenia kończy się uzyskaniem dwóch artefaktów: modelu akcji i klasyfikatora weryfikującego. W niektórych przypadkach klasyfikator weryfikujący dostarczony jest



Rysunek 4.1: Schemat procesu weryfikacji reguł akcji i rekomendacji

zewnątrznie i nie bierze udziału w fazie uczenia. Faza ta determinuje postać modelu akcji, reszta procesu służy tylko ocenieniu skuteczności wytworzonego modelu.

Po zakończonej fazie uczenia następuje faza testowania, na którą składają się trzy kroki:

- Testowanie klasyfikatora  $\Psi$  na zbiorze testowym. Uzyskanie wskaźników skuteczności klasyfikacji na zbiorze testowym jest niezbędne aby ustalić poziom odniesienia, do którego przyrównywana będzie ocena skuteczności modelu akcji. Jest to punkt kontrolny całego procesu oceny, gdyż wysokiej jakości klasyfikator jest kluczowy dla uzyskania wiarygodnych wyników.
- Transformacja przykładów. Przykłady ze zbioru treningowego zostaną zmodyfikowane tak, jakby zastosowane na nich zostały odpowiednio wybrane akcje z modelu, co stanowi symulację rzeczywistego wpływu na przykłady.
- Klasyfikacja. Zmodyfikowane we wcześniejszej fazie przykłady zostają podane na wejście klasyfikatora weryfikującego. Jeśli zmodyfikowany przykład zostanie rozpoznany jako należący do klasy docelowej, podczas gdy przed transformacją należał do klasy źródłowej, oznacza to, że model akcji zadziałał zgodnie z oczekiwaniem. Zagregowane wyniki klasyfikacji wszystkich przykładów posłużą do wyznaczenia wskaźników jakości modelu akcji.

Przedstawiony pokrótce ogólny schemat działania wymaga uszczegółowienia, uмотywowania oraz ostatecznie zdefiniowania pożądaných wskaźników jakości oraz ich interpretacji. Poszczególne kroki procesu zostaną opisane w detalach w dalszej części rozdziału.

### **4.3.1 Uczenie modelu akcji. Ważność klasyfikatora weryfikującego.**

Aby ocenić skuteczność modelu akcji, przede wszystkim niezbędne jest wyznaczenie dwóch zbiorów przykładów: treningowego i testowego. Zbiór testowy nie bierze udziału w fazie uczenia, a więc pozostaje nieznanym zarówno dla utworzonego modelu akcji, jak i dla klasyfikatora  $\Psi$  weryfikującego skuteczność. Zbiór treningowy i parametry zadane przez użytkownika, takie jak miara oceny jakości używana do nadzorowania procesu wzrostu i przycinania reguł akcji, wariant metody (w przód, wstecz) czy minimalne pokrycie reguły akcji, determinują testowany w dalszej fazie model akcji.

Wybór klasyfikatora weryfikującego  $\Psi$  również leży w gestii użytkownika. Wybór ten powinien być uzależniony od rozwiązywanego problemu, a wybrany sposób klasyfikacji powinien charakteryzować się wysoką skutecznością. Im lepszy będzie wybrany klasyfikator, tym bardziej wiarygodne będą wyniki oceny jakości modelu akcji. Na rysunku 4.1 pokazano, że klasyfikator  $\Psi$  powstaje w wyniku fazy uczenia na zbiorze treningowym, jednak nie ma przeszkód, by użyć gotowego klasyfikatora, którego skuteczność została już potwierdzona wcześniej. Proponowany proces wiąże ocenę modelu akcji z jakością klasyfikatora użytego do wyznaczenia tej oceny, więc niezależnie od źródła klasyfikatora, zostanie on poddany ocenie na zbiorze testowym.

Wykorzystując podany sposób oceny modeli akcyjnych należy zadbać o ustalenie wartości parametrów uczenia klasyfikatora weryfikującego i pozostałych faz procesu weryfikacji. Pomiedzy kolejnymi wykonaniami można tylko zmieniać parametry indukcji modelu akcji. Jeśli parametry pozostałych procesów nie zostaną zachowane, niemożliwe będzie zestawianie wyników dla różnych modeli akcji.

Ważność wpływu jakości klasyfikatora weryfikującego na interpretację uzyskanej oceny skuteczności modeli akcji wymaga najwyższej uwagi. Klasyfikator weryfikujący zostanie wykorzystany jako źródło wiedzy na temat tego, jak bardzo przykłady poddane transformacji przez model akcji przypominają przykłady klasy docelowej. Zastosowanie nieskutecznego (lub – nie dość skutecznego) klasyfikatora w tej roli nie pozwoli na wyciągnięcie właściwych wniosków z procesu oceny modelu akcyjnego. Przykładowo, testując pewien model akcyjny może dojść do sytuacji, w której wszystkie przykłady poddane transformacji zostały przez klasyfikator weryfikujący rozpoznane jako należące do klasy docelowej. Jeśli wcześniej klasyfikator weryfikujący wykazał się wysoką skutecznością zarówno na zbiorze treningowym i testowym, można przyjąć że ocena działania modelu akcyjnego jest wiarygodna. Jeśli jednak klasyfikator uzyskał dobry wynik na zbiorze treningowym, a kiepski na testowym, to ocena modelu akcyjnego nie jest wiarygodna. Należy wybrać inny klasyfikator weryfikujący, lub skorygować jego parametry uczenia, rozmiary zbiorów treningowego i testowego. Przeprowadzenie zaproponowanego procesu testowania modeli akcyjnych z użyciem niedostatecznie dobrego klasyfikatora spowoduje albo nieprawdziwie niskiej oceny modelu akcyjnego albo - co gorsze - nieprawdziwie wysokiej oceny.

Jak widać, klasyfikator weryfikujący jest kluczowym elementem całego procesu, gdyż pozwala stwierdzić, jak bardzo odpowiedzi modelu akcji są skuteczne, a na ile przypominają one dzieło przypadku. Klasyfikator  $\Psi$  pełni w procesie niejako rolę sędziego, lub źródła prawdy na temat tego do jakiej klasy przynależy dany przykład.

Nic nie stoi na przeszkodzie by w tę rolę wszedł prawdziwy ekspert domenowy, choć oczywiście znacznie wydłuży to czas trwania całego procesu oceny. Rozwiązaniem łączącym sformalizowaną wiedzę ekspercką z zaletami systemu uczącego się i szybkością wykonania mogłoby w takiej sytuacji być zastosowanie systemów reguł eksperckich [39].

### **4.3.2 Transformacja przykładów. Symulowanie zastosowania akcji.**

Jak wspomniano wcześniej, jedynym pewnym sposobem zweryfikowania czy akcja przynosi pożądany efekt, jest jej wdrożenie w życie oraz obserwacja losów przykładów. Sposób oceny modelu akcji opiera się na zastąpieniu modyfikacji fizycznych obiektów poprzez symulację takiego działania, które polegać będzie na zmianie wartości atrybutów analizowanych przykładów zgodnie z sugestią reprezentowaną przez odpowiednią akcję. Innymi słowy, symulowane będzie wykonanie akcji, bez ponoszenia kosztów i konsekwencji wykonywania ich w rzeczywistości. Działanie takie będzie nazywane transformowaniem przykładów przy użyciu akcji.

Transformowanie przykładów wydaje się być oczywistą operacją, i faktycznie tak jest, ale tylko dla zbiorów danych które są opisane w pełni przy użyciu atrybutów nominalnych. Wówczas również wszystkie akcje w modelu akcji wskazują konkretne wartości atrybutów, które należy ustawić, aby przetransformować przykład. Sprawa komplikuje się w przypadku atrybutów numerycznych. Przedstawione w niniejszej pracy modele akcji nie zwracają konkretnych wartości dla atrybutów numerycznych, ale przedziały wartości, do których należy „przenieść” przykład. Proces transformacji musi zapewnić, że przykład poddany takiemu procesowi nie będzie się różnić formą od przykładu oryginalnego, tj. że zachowane zostaną liczba i rodzaj atrybutów opisujących przykład oraz ich dziedziny. Krótko mówiąc, w przypadku atrybutów numerycznych nie da się wprost wywieść docelowej wartości na podstawie samej akcji i wartości źródłowej. Aby móc wykorzystać symulowane transformowanie przykładów, niezbędne jest podanie metody uzyskiwania dokładnych wartości na podstawie zakresu znalezionej w części docelowej akcji elementarnej opartej o atrybut numeryczny.

Podanie takiej metody nie nastęrcza problemów dla atrybutów nominalnych, przedstawia ją poniższy przykład:

**Przykład 4.3.1** *W pewnym zbiorze danych występuje następujący przykład: ((wilgotność, niska), (temperatura, 13.0)). Po przetransformowaniu go akcją postaci (wilgotność, niska → wysoka) uzyskamy nowy przykład: ((wilgotność, wysoka), (temperatura, 13.0))*

Sprawa komplikuje się w przypadku atrybutów numerycznych, gdzie akcja zawiera przedział wartości. Którą z wartości z tego przedziału wybrać? Jest to pytanie ważne nie tylko w kontekście testowania modeli akcji i symulowania zastosowania akcji, ale odpowiedź na nie może być także użyteczna na etapie wdrażania akcji w życie na obiektach rzeczywistych. W tej drugiej sytuacji mogą jednak występować dodatkowe ograniczenia związane z kosztem, procesem technologicznym i jego obostrzeniami czy w ogóle wykonalnością pewnych akcji. Nie mniej, aby zrealizować proces oceny modelu akcji, sposób uzyskiwania konkretnych wartości na podstawie akcji i wartości źródłowej trzeba podać. W zastosowaniach szczególnych warto sposób ten dostosować do rozwiązywanego problemu i uwzględnić istniejące ograniczenia lub znane preferencje dotyczące wyboru konkretnych wartości. Można podać parę sposobów, które są niezależne od rozwiązywanego problemu:

- Przyjmując, że każda wartość z zadanego przedziału jest równie dobra, wartość docelową można wylosować z równym prawdopodobieństwem.
- Innym podejściem jest wybranie wartości środkowej.
- Na zadany przedziale można zbudować rozkład prawdopodobieństwa, w którym wartości bliżej środka są bardziej prawdopodobne, i wylosować wartość z takiego rozkładu.

Sposób wyboru wartości docelowej można też uzależnić od częstości występowania tej wartości w zbiorze treningowym, i jest to metoda polecana przez autora. Ideą która tu przyświeca jest chęć sugerowania wartości, które były już zaobserwowane w populacji przykładów z klasy docelowej. Aby wyznaczyć wartość docelową, najpierw trzeba wyszukać wszystkie przykłady klasy docelowej ze zbioru treningowego, które są pokrywane przez część docelową analizowanej akcji. Następnie wyznacza się medianę wartości występującą w tym zbiorze przykładów na atrybucie modyfikowanym przez analizowaną akcję. Mediana ta przyjmowana jest jako wartość docelowa i wpisywana jako wartość atrybutu w transformowanym przykładzie.



**Przykład 4.3.2** Przyjmijmy następujący zbiór danych opisujący zarejestrowane odczucia pewnych osób na warunki pogodowe:

<i>id</i>	<i>temperatura</i>	<i>wilgotność</i>	<i>odczucie</i>
1	13.0	niska	przyjemne
2	14.0	wysoka	przyjemne
3	27.0	wysoka	nieprzyjemne
4	27.0	niska	przyjemne
5	35.5	wysoka	nieprzyjemne
6	34.0	niska	nieprzyjemne
7	-3	wysoka	przyjemne
8	-8.0	wysoka	nieprzyjemne

Klasa „nieprzyjemne” będzie pełnić rolę klasy źródłowej, a „przyjemne” klasy docelowej. Przyjmijmy, że chcemy zasymulować zastosowanie akcji ( $temperatura, (30.0, \infty) \rightarrow (10.0, 28.0)$ ) na przykładzie o *id* 6. Do wyznaczenia wartości docelowej bierzemy wartości atrybutu *temperatura* przykładów o *id* 1, 2, 4, ponieważ są pokrywane przez część docelową akcji oraz należą do klasy docelowej. Mediana tych wartości wynosi 14 i zostanie użyta do zmodyfikowania przykładu.

W praktyce można by zastosować dowolną inną średnią, na przykład arytmetyczną, jednak zastosowanie mediany pozwala uzyskać odporność na wartości mocno odstające, co pozwala zignorować ewentualne anomalie występujące w zbiorze treningowym.

Istotnym krokiem całego procesu, który należy wykonać przed transformacją przykładów, jest wyłonienie w zbiorze testowym przykładów klasy źródłowej. Przykłady klasy docelowej nie będą brać udziału w ustalaniu skuteczności modelu akcji. Dzieje się tak, ponieważ po pierwsze celem utworzenia modelu akcji nie jest stosowanie go potem na obiektach, których nie chcemy zmieniać, więc odpowiedź takiego modelu na przykład już należący do klasy docelowej nie ma żadnej wartości. Po drugie, jeśli model został wyuczony tylko przejść z klasy źródłowej do docelowej, to po prostu nie będzie w stanie wygenerować dobrej odpowiedzi na przykłady oryginalnie należące do klasy docelowej. Uwzględnienie oceny skuteczności transformacji takich przykładów w całkowitym wyniku przekreśliłoby sens badania skuteczności takiego modelu.

Ostatecznie odfiltrowane przykłady klasy docelowej zostają przetransformowane zgodnie z akcjami podanymi przez model akcji. Tak zmodyfikowany zbiór przykładów zostaje przekazany do ostatniej fazy testowania, tj. do klasyfikacji przez klasyfikator weryfikujący.

### 4.3.3 Klasyfikacja i testowanie klasyfikatora. Metryki modelu akcji.

Zanim zbiór zmodyfikowanych przykładów zostanie poddany klasyfikacji, należy ostatecznie ustalić poziom skuteczności rozpoznawania przykładów poszczególnych klas przez  $\Psi$ . W poprzednich sekcjach opisano już dlaczego niezbędna jest wysoka skuteczność tego klasyfikatora, i jak wpływa ona na interpretacje wyników całego procesu.

Do przetestowania klasyfikatora wykorzystywany jest zbiór testowy w niezmienionej formie, zawierający zarówno przykłady klasy docelowej jak i źródłowej, pomimo tego, że do transformacji przekazywane są tylko przykłady klasy źródłowej. Nie możemy jednak zastosować tej samej motywacji przy wyborze przykładów, jaką zastosowano przy wyborze przykładów do transformacji. Klasyfikator jest niezależny od modelu akcji i musi być sprawny zarówno w rozpoznawaniu przykładów klasy docelowej jak i źródłowej. Tylko taki klasyfikator jest w stanie potwierdzić (lub obalić) skuteczność badanego modelu akcji. Może się przecież zdarzyć, że model akcji nie będzie skutecznie transformować przykładów. Gdybyśmy użyli klasyfikatora wytrenowanego tylko na przykładach klasy docelowej, nie byłby on w stanie zgłosić błędu transformacji, lub co gorsza - przykład dowolnej postaci byłby przez niego uznawany za przykład klasy docelowej. Z tych względów wybór i ocena klasyfikatora gra kluczową rolę w całym procesie. Używając tej metody warto ustawić próg minimalnej dopuszczalnej skuteczności klasyfikatora na zbiorze testowym.

Jeśli klasyfikator zostanie uznany za wystarczająco dobry, można wreszcie przedstawić mu zmodyfikowane przykłady. W idealnej sytuacji, w której testujemy sprawny model akcji używając dobrze wytrenowanego klasyfikatora, wszystkie przykłady zostaną przyporządkowane do klasy docelowej. W rzeczywistości będzie to jednak sytuacja rzadka. Do określenia jak bliski ideałowi jest badany model akcji może posłużyć zestaw metryk, określających skuteczność jego działania. W dalszej części zaproponowano kilka takich metryk, które pozwalają monitorować zachowanie modelu akcji.

Po pierwsze, może się zdarzyć, że model akcji nie wygeneruje żadnej rekomendacji dla pewnej liczby przykładów. Ta sytuacja jest szczególnie możliwa gdy badanym modelem jest zbiór reguł akcji - jeśli będziemy próbowali transformować przykłady, których nie pokrywa żadna z dostępnych reguł akcji, przykłady te po transformacji będą tożsame z przykładami początkowymi. Nie przekreśla to jeszcze zupełnie

badanego modelu, natomiast częstość występowania takiej sytuacji niesie istotną informację o przydatności tego modelu. Można teraz zdefiniować pierwszą metrykę oceny modeli akcji, jaką da się uzyskać w opisywanym procesie testowania. Będzie nią stosunek liczby przykładów źródłowych, dla których nie wygenerowano akcji do wszystkich przykładów źródłowych w zbiorze testowym. Metrykę tę oznaczajmy  $m_1$  (4.4).

$$m_1 = \frac{|\{\text{Przykłady pozostawione bez akcji}\}|}{|\{\text{Przykłady źródłowe w zbiorze testowym}\}|} \quad (4.4)$$

Metryka ta pozwala ocenić, na ile ogólny jest badany model akcji, tj. jak dobrze jest w stanie wskazywać sugestie zmian dla przykładów, których wcześniej nie znał. Oczywiście model jest bardziej ogólny, a tym samym lepszy, gdy wartość  $m_1$  zmierza do zera. Zakres wartości tej miary to  $[0, 1]$ . W najgorszym przypadku, gdy do żadnego z przykładów źródłowych w zbiorze testowym nie zostanie dopasowana akcja, wartość miary wyniesie 1. Wynika to z faktu, że transformacji poddawane są tylko przykłady źródłowe.

Wydaje się, że najistotniejszą informacją o modelu akcji jest to, czy wygenerowany zestaw akcji dla danego przykładu sprawia, że przykład ten przejdzie do klasy docelowej. W procesie badania modelu tę informację uzyskujemy, podając zmodyfikowane przykłady na wejście klasyfikatora  $\Psi$  i monitorując jego odpowiedź. Jeśli wynikiem jest rozpoznanie przykładu jako należącego do klasy docelowej, uznajemy to za sukces, przyjmując przy tym że zaproponowany zestaw akcji jest skuteczny. Skuteczność modelu akcji możemy więc liczbowo wyrazić jako stosunek liczby przykładów zakwalifikowanych przez  $\Psi$  do klasy docelowej do liczby przykładów, dla których wygenerowano akcje.

$$m_2 = \frac{|\{\text{Przykłady rozpoznane jako docelowe}\}|}{|\{\text{Przykłady, dla których wygenerowano akcje}\}|} \quad (4.5)$$

Metryka ta pozwala oszacować skuteczność modelu akcji, a raczej sugerowanych przez ten model akcji poprzez określenie dla ilu przykładów podane akcje były skuteczne. Nie jest to oszacowanie dokładne, gdyż nie uwzględnia ono potencjalnych błędów popełnianych przez klasyfikator  $\Psi$ , ale cały proces oceny opiera się o zaufanie do tego klasyfikatora, korzystając z niego należy być świadomym, że uzyskane wyniki są orientacyjne. Metryka podana w 4.5 jest metryką korzyści - im wyższa jej wartość, tym lepiej. Można zdefiniować komplementarną do niej metrykę kosztu:

$$m_3 = \frac{|\{\text{Przykłady rozpoznane jako źródłowe}\}|}{|\{\text{Przykłady, dla których wygenerowano akcje}\}|} \quad (4.6)$$

Metryka  $m_3$  pokazuje, jak bardzo model myli się w podawanych akcjach. Dobry model będzie charakteryzować się wartościami bliskimi zero. Zakres wartości metryk  $m_2$  i  $m_3$  jest równy  $[0, 1]$ . Na wejście klasyfikatora podawane są tylko przykłady, które przeszły transformację.

Podobnie jak przy ocenie zdolności klasyfikacyjnych, nie zawsze będzie nas interesować uzyskanie możliwych najlepszych wyników wszystkich metryk. W niektórych sytuacjach model akcji, który ma względnie wysoką wartość  $m_1$ , ale również wysoką wartość  $m_2$  będzie akceptowalny, gdyż choć nie jest bardzo ogólny i nie zawsze zadziała, to gdy już zadziała, podaje dobre sugestie, co może stanowić wartość dla użytkownika metody. Raczej nieakceptowalne będą modele, dla których wartość  $m_1$  jest wysoka przy jednocześnie niskiej wartości  $m_2$ . Do tego trzeba zawsze odnieść uzyskane wyniki do jakości klasyfikatora, który był wykorzystany do oceny modelu akcji, jak już to omówiono we wcześniejszych paragrafach.

#### 4.3.4 Rekomendacje a problemy regresyjne

W rozdziale 3 obok sposobu indukcji reguł akcji w zadaniu klasyfikacji opisano także metodę uzyskiwania reguł akcji dla zbiorów opisujących problemy regresyjne, tj. takie, w których zmienna decyzyjna jest atrybutem typu ciągłego, numerycznego. Bieżący rozdział wprowadza metodę uzyskiwania rekomendacji oraz sposób oceny skuteczności modelu akcji dla problemu klasyfikacji. Wprowadzając kilka modyfikacji w opisanych już rozwiązaniach można je dostosować do działania na zbiorach regresyjnych. Zmiany te zostaną opisane w niniejszej części.

Przede wszystkim trzeba zauważyć, że definicja 4.2.1 obowiązuje zarówno dla reguł akcji klasyfikacyjnych jak i regresyjnych, gdyż nie wymusza ona konkretnej formy atrybutu decyzyjnego. W szczególności można przyjąć, że reguły akcji regresyjne i klasyfikacyjne różnią się tym, że wartość w konkluzjach tych reguł jest elementem zbioru dyskretnego w przypadku zadania klasyfikacji, oraz wartością rzeczywistą gdy mowa o zadaniu regresji. Dalsze różnice, takie jak wyróżnienie pewnej wartości jako docelowej w przypadku klasyfikacji czy zmienność zbioru przykładów pozytywnych reguły regresyjnej (por. definicja 7), nie mają wpływu na rozumienie do czego prowadzi zastosowanie reguły akcji. Dzięki potraktowaniu konkluzji reguły akcji jako wskazania na zmianę jednej wartości na inną i jednoczesnym zaniedbaniu postaci

zbioru, z którego te wartości pochodzą, można bez większych problemów zastosować metody przedstawione dla problemu klasyfikacji w zadaniu regresji.

Jednym z przejawów tej korzyści jest fakt, że budując algorytm indukcji rekomendacji dla problemu regresji, można skorzystać z meta-tablicy o dokładnie takiej samej postaci jak opisano w rozdziale 4.2.1. Konstrukcja meta-tablicy nie zależy od postaci konkluzji reguł, z których została zbudowana. Kluczowe jest zbudowanie meta-przykładów o jak najdokładniejszych przedziałach wartości i odniesienie ich do wartości konkluzji, przy czym nieważna jest jej postać ani inne meta-dane, które są istotne gdy konkluzja jest częścią reguły. Celem utworzenia meta-tablicy jest stworzenie bazy danych na temat charakterystyk (wartości atrybutów) przykładów skojarzonych z wartościami wyróżnionego atrybutu decyzyjnego.

Informacje o tym, jakie znaczenie ma wartość konkluzji w meta-tablicy jest jednak istotne w trakcie przeszukiwania meta-tablicy podczas budowania rekomendacji. Dla problemu klasyfikacji niezbędna jest wiedza o tym, która klasa jest źródłowa, a która docelowa, zaś w zadaniu regresji nie ma w ogóle podziału na klasy, za to istotną cechą uzyskanych rekomendacji może być to, by sugerowały takie zmiany, które pozwolą uzyskać wartość atrybutu decyzyjnego wyższą lub niższą od przypisanej do analizowanego przykładu.

Proces weryfikacji reguł akcji w zadaniu regresji wymaga pewnych zmian, nie wpływających jednak na jego strukturę ogólną. Przede wszystkim, w eksperymencie opisanym w rozdziale 4.3 kluczowe jest dobranie odpowiedniego klasyfikatora weryfikującego  $\Psi$ . Z oczywistych względów, dla problemu regresji należy tutaj użyć innego algorytmu, który pozwala przewidywać wartości numerycznych atrybutów decyzyjnych jak np. algorytm XGBoost, maszyny wektorów podpierających (SVM) czy wielowarstwowe perceptrony (MLP). Użycie programu weryfikującego będzie takie samo jak w przypadku klasyfikacji, i tak samo ocena jakości zastosowanego programu będzie istotna przy interpretacji wyników całego eksperymentu. Proces testowania modelu akcyjnego w przypadku regresji niewiele różni się od procesu przedstawionego na rysunku 4.1 dla klasyfikacji. Zgodnie z podanym opisem, wystarczy wymienić w tym schemacie „Klasyfikator  $\Psi$ ” na odpowiedni algorytm właściwy dla regresji, oraz fazę klasyfikacji zastąpić fazą predykcji wartości ciągłej. Oczywiście, zmienia się też sposób indukcji modelu akcji zawarty w fazie uczenia, gdyż trzeba zastosować program indukcji reguł lub rekomendacji odpowiedni dla regresji. Pewne zmiany muszą zostać także wprowadzone w fazie transformacji przykładów. Przede wszystkim, do fazy transformacji wchodzi wszystkie przykłady zbioru testowego. Nie

zmienia się sposób mutacji atrybutów nominalnych - wartość przykłady na danym atrybucie zostaje ustawiona na wartość sugerowaną przez akcję podaną w regule lub rekomendacji. W przypadku atrybutów numerycznych, nadal wybierana jest mediana wartości zarejestrowanych dla wszystkich przykładów w zbiorze treningowym, które zostały pokryte przez część docelową właściwej dla danego atrybutu akcji elementarnej. Przypomnijmy, że w przypadku problemu klasyfikacji, nałożono dodatkowy warunek, że muszą to być przykłady klasy docelowej.

W wyniku przeprowadzenia eksperymentu weryfikacyjnego dla problemu regresji uzyskany zostanie inny niż dla klasyfikacji zestaw danych opisujących ten proces, m.in. ze względu na brak definicji klasy źródłowej i docelowej, czy też w ogóle - brak klas - trudniej jest określić, czy mutacja przykładu była skuteczna. W dalszej części zdefiniowane zostaną trzy metryki umożliwiające ocenienie skuteczności danego regresyjnego modelu akcji.

Aby wprowadzić te metryki trzeba najpierw ustalić postać zbioru przetransformowanych przykładów. Zbiór ten składa się z przykładów o takiej samej liczbie i formie atrybutów co zbiór oryginalny. Różnica leży w wartościach atrybutów, które na podstawie modelu akcji zostały zmienione w celu wpłynięcia na wartość atrybutu decyzyjnego, która także została zmodyfikowana przez model akcji. Tak zmienione przykłady podawane są na wejście algorytmu potrafiącego przewidywać (estymować) wartość atrybutu decyzyjnego na podstawie wartości pozostałych atrybutów warunkowych. Po zakończeniu procesu predykcji, dla każdego przykładu posiadamy dwie wartości atrybutu decyzyjnego: wskazaną przez metodę użytą do weryfikacji, oraz wskazaną przez model akcji. Pierwsze dwie metryki opierają się o chęć dowiedzenia się, na ile wartości sugerowane przez model akcyjny różnią się od tych wskazanych przez metodę predykcyjną.

Pierwsza proponowana metryka to wartość pierwiastka błędu średniokwadratowego RMSE (ang. *Root Mean Square Error*) definiowana wzorem:

$$mr_1 = RMSE(v'_d, \bar{v}_d) = \sqrt{\frac{1}{n} \sum_{i=1}^n (v'_{d_i} - \bar{v}_{d_i})^2} \quad (4.7)$$

gdzie  $\bar{v}_d$  oznacza uporządkowany zbiór predykcji wartości atrybutu decyzyjnego, a  $v'_d$  to uporządkowany zbiór wartości atrybutu decyzyjnego sugerowanych przez model akcji i  $n$  wyraża liczbę przykładów w zmutowanym zbiorze. Wykorzystanie wskaźnika RMSE ułatwia interpretację wyników, ponieważ wartość tej metryki jest wyrażona w takiej samej skali, jak wartości atrybutu decyzyjnego w badanym zbiorze danych.

Z drugiej strony, metryka ta jest silnie podatna na wpływ wartości odstających, dlatego warto wprowadzić drugą, podobną metrykę, która opiera się o ideę średniego błędu bezwzględnego MAE (ang. *Mean Absolute Error*):

$$mr_2 = MAE(v'_d, \bar{v}_d) = \frac{1}{n} \sum_{i=1}^n |v'_{d_i} - \bar{v}_{d_i}| \quad (4.8)$$

Metryka ta jest mniej czuła na wartości odstające, a każdy cząstkowy błąd wnosi proporcjonalny wpływ do ostatecznej wartości metryki, w przeciwieństwie do RMSE. MAE podobnie jak RMSE wyrażona jest w tej samej skali co wartości atrybutu decyzyjnego, dzięki czemu jest łatwa w interpretacji. Obydwie metryki oparte o uśredniony błąd będą mieć wartości bliższe zeru dla lepszych modeli akcyjnych, przy czym należy pamiętać o konieczności odniesienia uzyskanych wyników do jakości użytej metody predykcji wartości ocenionej na zbiorze testowym.

Metryki  $mr_1$  i  $mr_2$  będą czasem nazywane błędem RMSE i MAE, gdy kontekst wypowiedzi nie będzie utrudniał zrozumienia znaczenia, do błędu między jakimi wartościami nazwy te się odnoszą. Należy jeszcze zauważyć, że obydwie te metryki biorą pod uwagę tylko te przykłady, które podległy przetransformowaniu; inaczej ujmując, nie uwzględniają one przykładów, dla których model akcji nie wydał żadnej rekomendacji.

Ostatnia proponowana metryka jest próbą zastąpienia łatwej w interpretacji metryki  $m_2$  zdefiniowanej dla klasyfikacyjnego modelu akcji, której wartość jest interpretowana jako liczba skutecznie przetransformowanych przykładów. Podobnie obrazową metrykę dla problemu regresji można zdefiniować przyjmując następujące założenie: przykład zostanie uznany za skutecznie przetransformowany, jeśli wartość atrybutu decyzyjnego przewidziana w fazie predykcji znajduje się w pewnym niewielkim otoczeniu wartości sugerowanej przez regułę lub rekomendację, która była użyta w transformacji przykładu. Wspomniane otoczenie można przyjąć następująco:

$$(v'_{d_i} - s(v'_d), v'_{d_i} + s(v'_d)) \quad (4.9)$$

gdzie przez  $s(v'_d)$  oznaczono odchylenie standardowe wszystkich sugerowanych wartości atrybutu decyzyjnego w zbiorze przykładów zmutowanych, a  $v_{d_i}$  oznacza wartość sugerowaną dla  $i$ -tego przykładu. Ostatecznie, trzecią metrykę oceny modeli akcji dla problemu regresji można zdefiniować jako:

$$mr_3 = \frac{|\{v_{d_i} \in (v'_{d_i} - s(v'_d), v'_{d_i} + s(v'_d))\}|}{|\text{Przykłady, dla których wygenerowano akcje}|} \quad (4.10)$$

czyli stosunek liczby przykładów trafnie zmutowanych do liczby wszystkich przykładów, dla których podjęto jakiegokolwiek akcje. W tym przypadku, im wyższa wartość metryki, tym lepiej. Dodatkowo, bazując na definicji trzeciej metryki, łatwo wprowadzić specjalizowane metryki, poprzez zmianę założenia dotyczącego uznania transformacji jako skutecznej.



## 5 Eksperymenty

W tym rozdziale opisano wyniki eksperymentów przeprowadzonych z zastosowaniem metod opisanych w pracy. Metody zostaną porównane poprzez uruchomienie kilku wariantów algorytmu indukcji reguł akcji, algorytmu uzyskiwania rekomendacji oraz ocenienie ich w schemacie eksperymentu opisanym w sekcji 4.3.

Analiza przedstawionych metod przeprowadzona jest pod kątem dwóch zagadnień. Po pierwsze, interesujące są charakterystyki uzyskanych zbiorów reguł akcji, które pozwalają określić przydatność metod indukcji reguł akcji w zadaniu eksploracji danych i odkrywania wiedzy. Drugim interesującym aspektem - rzadko podejmowanym w literaturze - jest skuteczność reguł akcji i rekomendacji. Przez skuteczność reguł akcji i rekomendacji należy rozumieć odpowiedź na pytanie „jak często wykonanie reguły akcji (lub rekomendacji) pozwala zaobserwować, że przykład rzeczywiście zmienił przyporządkowanie z klasy źródłowej do docelowej”.

Rozdział ten podzielony jest dwie zasadnicze części: osobno opisano eksperymenty dla zadania klasyfikacji i regresji. W ramach każdej z części omówione zostaną charakterystyki zbiorów danych, na których przeprowadzono badania, uzyskane zbiory reguł oraz wyniki uzyskane w zadaniu transformowania przykładów, wraz z analizą statystyczną.

Na samym początku omówione zostaną parametry algorytmów, które są wspólne w zadaniu regresji i klasyfikacji oraz sposób porównywania uzyskanych wyników przy użyciu testów statystycznych.

### 5.1 Parametry algorytmów

W rozdziałach 3.1 i 4 przedstawiono kilka wariantów metod pozwalających uzyskiwać reguły akcji i rekomendacje. Warianty algorytmu indukcji reguł akcji generowane są poprzez zmianę strategii poszczególnych etapów ogólnego algorytmu, takich jak: kierunek indukcji, miara oceny jakości użyta do nadzorowania indukcji reguł czy ograniczenia nałożone na zakresy pojawiające się w częściach źródłowej i docelowej

poszczególnych akcji elementarnych. Wybór wariantu algorytmu indukcji reguł akcji będzie mieć wpływ na działanie algorytmu odkrywania rekomendacji, gdyż zbiór reguł akcji stanowi dla niego dane wejściowe. Próbując wdrożyć przedstawione metody, można posłużyć się przedstawioną w podrozdziale 4.3 metodą pozwalającą oszacować, jak dobrze wybrana metoda spisuje się w zadaniu transformowania przykładów, tj. sugerowaniu akcji, które pozwolą danemu przykładowi zmienić przyporządkowanie z klasy źródłowej do klasy docelowej.

W celu znalezienia odpowiedzi na pytanie, która kombinacja parametrów algorytmu indukcji reguł akcji pozwala na uzyskanie modelu akcji o najlepszej skuteczności w transformowaniu przykładów, dla każdego ze zbiorów z tabeli 5.2 przeprowadzono szereg eksperymentów wg. schematu opisanego w podrozdziale 4.3 za pomocą 10-krotnej walidacji krzyżowej. Warianty metody indukcji reguł akcji, biorące udział w porównaniu, zostały zbudowane poprzez różnicowanie następujących cech algorytmu:

- miara oceny jakości użyta do nadzorowania indukcji,
- kierunek indukcji: w przód lub wstecz,
- ograniczenie dopuszczalnych przedziałów w akcji elementarnej.

W eksperymentach użyto następujących miar oceny jakości reguł klasyfikacyjnych: *C2*, *WeightedLaplace*, *InformationGain*, *Correlation*, *RSS*. Szerszy opis dotyczących tych miar zawarto w podrozdziale 2.2.1. Kierunek indukcji reguł akcji omówiono w części 3.2. W rozdziale 3.1 wspomniano o trzech strategiach budowania zbioru kandydujących docelowych warunków elementarnych, które określają dopuszczalne przedziały atrybutów ciągłych w odkrywanych akcjach elementarnych.

W dalszej części rozdziału przyjęto następujący sposób identyfikacji wariantu algorytmu:  $\{Metoda\}$ - $\{Nazwa\ miary\}$ - $\{Kierunek\ indukcji\}$ - $\{Dopuszczalne\ przedziały\}$ . Poszczególne części mogą przyjmować następujące wartości:

- $\{Metoda\}$  : *reguły*, *rekomendacje* - zastosowanie wprost reguł akcji do transformacji przykładów jest określone angielską nazwą *reguły*, zaś użycie algorytmu indukcji rekomendacji - skrótem *rekomendacje* pochodzącym od angielskiego *recommendations*,
- $\{Nazwa\ miary\}$  : *C2*, *WeightedLaplace*, *InformationGain*, *Correlation*, *RSS* - nazwy te wprost odpowiadają nazwom miar używanym wcześniej,

- *{Kierunek indukcji}* : *F, B* - gdzie *F* oznacza indukcje w przód (od ang. *Forward*), a *B* - wstecz (ang. *Backward*),
- *{Dopuszczalne przedziały}* : *all-ranges, exclusive-only, not-intersecting* - gdzie każda z nazw przełożona na j. polski identyfikuje jedną ze strategii wyboru dopuszczalnych przedziałów w akcjach elementarnych.

Na przykład, algorytm indukcji reguł w przód nadzorowany miarą *C2* z brakiem ograniczeń dopuszczalnych przedziałów w akcjach elementarnych nazywany będzie *rules-C2-F-all-ranges*. Zbędne człony nazwy, tj. gdy dana część nie jest istotna dla rozważanego tematu, będą pomijane. Tam, gdzie to praktyczne, składowe budujące nazwę wariantu będą wymieniane osobno, na przykład w podsumowaniach tabelarycznych.

Przeprowadzone eksperymenty są realizacją schematu przedstawionego w rozdziale 4.3. W ramach schematu eksperymentu dokonuje się transformacji przykładów ze zbioru testowego. Jak opisano w części 4.1, inaczej postępuje się gdy modelem akcji jest zbiór reguł akcji, a inaczej gdy model ten stanowi algorytm indukcji rekomendacji. Algorytm rekomendacyjny w przedstawionych eksperymentach jest zbudowany w oparciu o zbiory reguł akcji wyindukowanych algorytmem scharakteryzowanym przez miarę, kierunek indukcji oraz strategię wyborów przedziałów zawartą w nazwie wariantu. Proces indukcji pojedynczej rekomendacji jest również nadzorowany tą samą miarą, która jest użyta do indukcji reguł akcji.

Reguły akcji mają istotny wpływ na działanie algorytmu rekomendacji, ponieważ stanowią dane wejściowe, na podstawie których rekomendacje są indukowane. Jak można odczytać z przedstawionych w niniejszej pracy pseudokodów, proces indukcji reguł jest sterowany także innymi parametrami, które nie będą podlegać zmianom w ramach przedstawionych wariantów algorytmu. Wartości tych parametrów zostały dobrane arbitralnie i są stałe, zarówno w eksperymentach dotyczących problemu klasyfikacji jak i regresji. Parametry te i ich wartości to:

- *mincov = 5*,
- liczba przykładów, które mogą zostać niepokryte został ustalony na 5% liczby przykładów odpowiedniej klasy (źródłowej lub docelowej, zależnie od wariantu) w zbiorze treningowym,
- opcjonalny krok przycinania reguł jest wykonywany w każdym z wariantów,

- wartości brakujące są ignorowane, tzn. nie biorą udziału w generowaniu warunków elementarnych, a przykład z brakującą wartością pewnego atrybutu nie będzie pokryty przez żaden warunek zbudowany na tym atrybucie. Taka strategia obsługi wartości brakujących została wybrana na podstawie wyników przeglądowej pracy Wohlraba i Fürnkranza [102],
- nazwa miary jakości w wariancie metody określa miarę nadzorującą zarówno wzrost jak i przycinanie reguł.

W schemacie eksperymentu, którego realizacja jest opisywana w tym rozdziale, istotną rolę odgrywa klasyfikator wykorzystany do weryfikowania skuteczności reguł akcji. Klasyfikatorem weryfikującym wykorzystanym w procesie testowania modeli akcyjnych jest XGBoost [13] (implementacja w języku R [71], z pakietu *xgboost* [14]). Wykorzystano domyślne parametry algorytmu w zakresie boostingu, przy maksymalnie 10000 rund, włączonej funkcji wczesnego stopu, jeśli skuteczność klasyfikatora nie poprawi się przez 10 kolejnych rund (*early stopping*) oraz funkcji celu ustalonej na regresję logistyczną dla klasyfikacji (*binary:logistic*).

W istniejących pracach dotyczących odkrywania reguł akcji, które opisują zastosowanie odkrytych reguł w ujęciu jednego lub maksymalnie kilku, podobnych problemów, nieco uwagi poświęca się prawidłowemu wyborowi atrybutów oznaczonych jako stałe i zmienne. Choć przedstawione w tej pracy algorytmy są w stanie przyjąć listę atrybutów stałych i odpowiednio uwzględnić jej zawartość w procesie indukcji reguł akcji, w ramach przeprowadzonych eksperymentów wszystkie atrybuty zostały oznaczone jako zmienne. Wynika to przede wszystkim z dużej różnorodności zbiorów testowych, dla których nie da się podać takiej samej listy atrybutów stałych, ale też faktu, że analiza algorytmów z dodatkowymi ograniczeniami spowodowałaby dalsze skomplikowanie opisu eksperymentów jednocześnie nie wnosząc wiele informacji o wynikach działania algorytmów, którą można by traktować jako uniwersalną, w oderwaniu od pojedynczego zbioru. Tematyka podziału zbioru atrybutów na stałe i zmienne nie jest także w sposób wyczerpujący opisana w niniejszej rozprawie oraz prowadzone badania nie skupiały się na analizie wpływu takiego podziału na działanie algorytmów.

Na koniec należy jeszcze wspomnieć o sposobie podziału zbiorów danych na zbiory treningowe i testowe. Eksperymenty przeprowadzane są w schemacie dziesięciokrotnej walidacji krzyżowej. Zbiór testowy wydzielany jest z oryginalnego zbioru danych. 80% przykładów wydzielonych z oryginalnego zbioru stanowi zbiór treningowy w

danym przebiegu, a pozostałych 20% staje się zbiorem testowym. Podział przykładów odbywa się w procesie losowania warstwowego (stratyfikacji), czyli tak, aby w każdym z utworzonych zbiorów zachowany został rozkład klas ze zbioru oryginalnego. Proces próbkowania wykonano przy użyciu klasy *StratifiedPartitionBuilder* z pakietu RapidMiner [60, 61].

## 5.2 Testy statystyczne w ocenie jakości algorytmów

W swojej pracy Demšar [22] proponuje, by do porównywania ze sobą wyników dwóch klasyfikatorów uzyskanych na wielu zbiorach danych stosować nieparametryczny test rang Wilcoxona [101]. Test ten ma tę przewagę nad – wydawałoby się naturalnym w przypadku porównywania dwóch zmiennych pomiarowych – testem t-Studenta, że nie zakłada normalności rozkładu zmiennych pomiarowych. Dodatkowo wartości odstające mają mniejszy wpływ na wynik testu Wilcoxona niż t-Studenta.

W przypadku porównywania ze sobą wyników więcej niż dwóch algorytmów zaleca się stosowanie nieparametrycznego testu Friedmana [29, 30] do sprawdzenia hipotezy zerowej mówiącej o tym, że wszystkie algorytmy są sobie równe, a w przypadku jej odrzucenia należy przeprowadzić analizę *post hoc*, przy użyciu testu Nemenyi [63]. Wyniki takiej analizy można zwizualizować w postaci CD-diagramów (czyli diagramów różnicy krytycznej, ang. *Critical Difference diagram*), które pokazują grupy algorytmów o statystycznie istotnie różnych rangach wyznaczonych w trakcie procedury Nemenyi.

Ze względu na mnogość wariantów algorytmów zaproponowanych w tej pracy, w dalszej części w głównej mierze posługiwać będziemy się testem Friedmana i CD-diagramami. Prześledźmy na przykładzie stosowanie testu Friedmana. Załóżmy  $k = 3$  algorytmy klasyfikacyjne, których skuteczność (przy czym nieistotne jest czy do określenia rangi użyto miary dokładności czy błędu) zbadano na  $N = 16$  zbiorach danych. Niech  $R_i^j$  oznacza rangę j-tego algorytmu w ramach i-tego zbioru danych. Jeśli wynik dla dwóch lub więcej algorytmów jest taki sam, przyznawana jest ranga będąca średnią. W tabeli 5.1 przedstawiono przykładowe wyniki dla trzech hipotetycznych algorytmów oraz przypisano odpowiednie rangi.

Test Friedmana porównuje średnie rangi  $R_j = \frac{1}{N} \sum_i r_i^j$  każdego z algorytmów. Hipoteza zerowa mówi, że *wszystkie algorytmy są takie same, a co za tym idzie ich rangi powinny być równe*. Dla tak zdefiniowanego testu, należy obliczyć statystykę Friedmana (wzór 5.1):

Numer zbioru	Algorytm A	Algorytm B	Algorytm C
1	74.71 (2)	73.51 (3)	88.60 (1)
2	72.95 (3)	76.51 (2)	93.38 (1)
3	81.98 (1)	77.52 (3)	79.73 (2)
4	77.39 (1)	70.48 (3)	76.20 (2)
5	77.50 (2)	75.27 (3)	83.37 (1)
6	70.18 (3)	73.01 (2)	80.88 (1)
7	74.71 (3)	81.03 (2)	81.23 (1)
8	71.41 (3)	73.61 (2)	82.54 (1)
9	78.70 (1)	78.01 (2)	72.86 (3)
10	75.80 (3)	80.60 (1)	80.59 (2)
11	67.04 (3)	78.66 (2)	89.19 (1)
12	72.39 (3)	73.14 (2)	76.16 (1)
13	71.27 (3)	72.68 (2)	87.54 (1)
14	72.02 (3)	80.53 (1)	80.19 (2)
15	85.72 (1.5)	76.20 (3)	85.72 (1.5)
16	80.04 (2)	78.13 (3)	84.40 (1)

**Tabela 5.1:** Przykładowe wyniki uzyskane przez trzy algorytmy na testowych zbiorach danych wraz z przypisanymi rangami (w nawiasach).

$$\chi_F^2 = \frac{12N}{k(k+1)} \left[ \sum_j R_j^2 - \frac{k(k+1)^2}{4} \right] \quad (5.1)$$

która została jednak uznana za zbyt konserwatywną przez Imana i Davenporta [47], którzy zaproponowali inną statystykę:

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1) - \chi_F^2} \quad (5.2)$$

która podlega rozkładowi F z  $(k-1)$  i  $(k-1)(N-1)$  stopni swobody. Ta właśnie statystyka znajdzie zastosowanie w testach Friedmana przeprowadzanych w tej pracy. Jeśli dla danego przypadku wartość tej statystyki jest mniejsza niż wartość krytyczna odczytana z tabel statystycznych, to należy przyjąć hipotezę zerową mówiącą o równości algorytmów, przeciwnie możemy ją odrzucić i przeprowadzić test Nemenyi. W ramach tego testu, średnie rangi algorytmów są ze sobą parami porównywane. Dwa algorytmy możemy uznać za statystycznie istotnie różne, jeśli ich średnie rangi

różnią się od siebie co najmniej o wartość krytyczną (CD) wyznaczoną zgodnie ze wzorem:

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \quad (5.3)$$

gdzie  $q_\alpha$  jest wartością krytyczną pochodząca ze studentyzowanej statystyki rozstępu, o ustalonym poziomie istotności  $\alpha$  oraz jest zależna od ilości klasyfikatorów  $k$ , dodatkowo podzielona przez  $\sqrt{2}$ .

Kontynuując przykład oparty o dane zawarte w tabeli 5.1, można wyznaczyć średnie rangi poszczególnych algorytmów:

$$\begin{aligned} R_A &= \frac{1}{16}(2 + 3 + 1 + 1 + 2 + 3 + 3 + 3 + 1 + 3 + 3 + 3 + 3 + 3 + 1.5 + 2) = \\ & \qquad \qquad \qquad \frac{1}{16} \cdot 36.5 = 2.34375 \\ R_B &= \frac{1}{16}(3 + 2 + 3 + 3 + 3 + 2 + 2 + 2 + 2 + 1 + 2 + 2 + 2 + 1 + 3 + 3) = \\ & \qquad \qquad \qquad \frac{1}{16} \cdot 35 = 2.25 \\ R_C &= \frac{1}{16}(3 + 1 + 1 + 1 + 1 + 1 + 3 + 1 + 2 + 3 + 1 + 2 + 3 + 1 + 1.5 + 2) = \\ & \qquad \qquad \qquad \frac{1}{16} \cdot 21.5 = 1.40625 \end{aligned} \quad (5.4)$$

Następnie, możemy obliczyć wartość statystyki Friedmana:

$$\begin{aligned} \chi_F^2 &= \frac{12 \cdot 16}{3(3+1)} \cdot \left[ (2.34375^2 + 2.25^2 + 1.40625^2) - \frac{3(3+1)^2}{4} \right] = \\ & \qquad \qquad \frac{192}{12} \cdot \left[ (5.49316 + 5.0625 + 1.97753) - \frac{48}{4} \right] = \\ & \qquad \qquad 16 \cdot [12.533203 - 12] = 16 \cdot 0.533203 = 8.531248 \end{aligned} \quad (5.5)$$

Ostatecznie, należy wyznaczyć wartość statystyki F:

$$F_F = \frac{16 - 1 \cdot 8.531248}{16 \cdot (3 - 1) - (8.531248)} = \frac{127.96872}{32 - 8.531248} = \frac{127.96872}{23.468752} = 5.452727 \quad (5.6)$$

Teraz należy odszukać wartość krytyczną dla  $3 - 1 = 2$  i  $(3 - 1) \cdot (16 - 1) = 30$  stopni swobody. Z tablic statystycznych rozkładu F można odczytać, że dla poziomu istotności  $\alpha = 0.05$  wynosi ona 3.32. Ponieważ wyznaczona wartość statystyki  $F_F$  jest większa od tej wartości, możemy odrzucić hipotezę zerową mówiącą o tym, że

wszystkie badane algorytmy są sobie równe (tj. że średnio mają tę samą rangę). W dalszym ciągu można przeprowadzić test Nemenyi, który pozwoli wykazać różnice między algorytmami.

Test Nemenyi rozpoczyna się od wyznaczenia wartości krytycznej zgodnie ze wzorem 5.3. Dla  $\alpha = 0.05$  i  $k = 3$  algorytmów oraz  $N = 16$  zbiorów, wartość  $q_\alpha = 2.424$ . W związku z tym,  $CD = 2.424 \cdot \sqrt{\frac{3 \cdot (3+1)}{6 \cdot 16}} = 2.424 \cdot \sqrt{\frac{12}{96}} = 2.424 \cdot \sqrt{0.125} = 0.857$ . Dwa algorytmy różnią się od siebie istotnie, jeśli różnica między ich średnimi rangami jest wyższa od tej wartości. W prezentowanym przykładzie różnice te wynoszą:

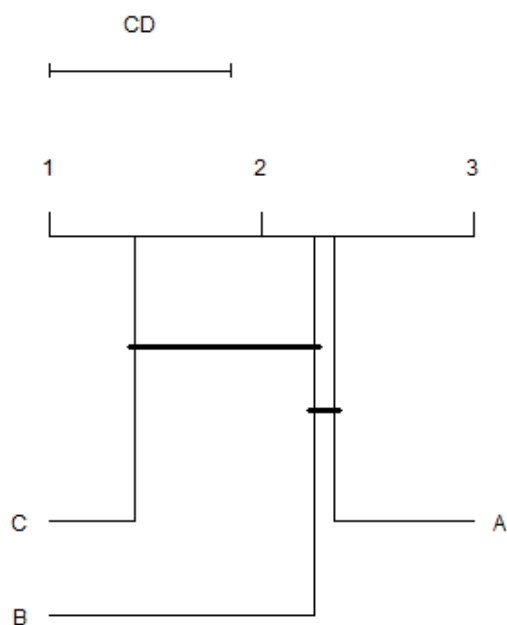
- między algorytmem A i B 0.09375
- między algorytmem A i C 0.9375
- między algorytmem B i C 0.84375

Na tej podstawie można powiedzieć, że algorytm C różni się istotnie od algorytmu A. Jednocześnie musimy stwierdzić, że na podstawie dostępnych danych, test Nemenyi nie pozwala rozstrzygnąć jednoznacznie o różnicach między algorytmem B i C oraz między algorytmami A i B. Nie jest to sytuacja rzadka - test Nemenyi uważany jest za mocno konserwatywny i często zdarza się, że choć test Friedmana pozwala odrzucić hipotezę zerową mówiącą o równości badanych algorytmów, test Nemenyi nie wykazuje istotnie różnych podgrup.

Przy większej liczbie porównywanych metod opisowe ujęcie wykrytych różnic między badanymi algorytmami zaczyna być kłopotliwe. Pomocną formą wizualizacji wyników testu Nemenyi są diagramy CD. Na rysunku 5.1 zaprezentowano diagram CD sporządzony dla omawianych przykładowych danych. W lewym górnym rogu graficznie przedstawiono wartość różnicy krytycznej CD. Algorytmy, których średnia ranga nie różni się o więcej niż o wartość CD od siebie, połączone są pogrubioną linią, tworząc grupę algorytmów. Średnie rangi poszczególnych algorytmów zaznaczone są graficznie na osi, na której wartości rang rosną od lewej do prawej. Wobec tego najlepszy algorytm (grupa najlepszych algorytmów) umieszczona będzie po lewej stronie diagramu. W oryginalnej propozycji Demšara oś jest odwrócona tak, aby lepsze metody znajdowały się po prawej stronie. Należy zwrócić uwagę na to, że jeśli w danej grupie algorytmów pewien algorytm znajduje się na CD-diagramie bliżej niższej rangi, oznacza to że częściej zajmował on tę pozycję niż pozostałe algorytmy w grupie, tzn. że w ramach grupy wzajemne położenie algorytmów ma znaczenie.

W ostatnich latach do porównań wielu klasyfikatorów zaleca się także stosowanie testu Wilcoxon z korektą p-wartości dla porównań wielokrotnych [91], wymaga on





**Rysunek 5.1:** Diagram CD sporządzony dla danych z tabeli 5.1 dla poziomu istotności  $\alpha = 0.05$ .

jednak przeprowadzenia testu dla każdej pary konkurujących algorytmów. Ze względu na mnogość analizowanych wariantów, oraz intuicyjne, obrazowe przedstawianie wyników poprzez CD diagramy, w niniejszej pracy zdecydowano się na wykorzystanie testu Friedmana z analizą post-hoc Nemenyi.

## 5.3 Reguły akcji w zadaniu analizy danych klasyfikacyjnych

W tym rozdziale przedstawiono i omówiono uzyskane zbiory reguł i wyniki procesu weryfikacji modeli akcyjnych zbudowanych ze zbiorów reguł akcji i rekomendacji w zadaniu analizy danych klasyfikacyjnych.

### 5.3.1 Charakterystyka zbiorów danych

Dla potrzeb eksperymentów przygotowano listę zbiorów danych pochodzących z repozytorium UCI [23]. Lista przedstawiona w tabeli 5.2 zawiera zbiory, których zmienna decyzyjna jest typu wyliczeniowego, dzięki czemu są one odpowiednie do testów w zadaniu klasyfikacji. Oprócz nazwy zbiorów przedstawiono ich podstawowe charakterystyki, takie jak liczba przykładów i atrybutów (w tym atrybutów nominalnych i numerycznych), jak i oznaczenie, która wartość atrybutu decyzyjnego została w eksperymentach uznana za klasę źródłową, a która za docelową. Na końcu tabeli umieszczono informację o dokładności klasyfikacji, jaką uzyskano stosując klasyfikator XGBoost na zbiorze testowym i treningowym. Są to wartości uśrednione dla 10 przebiegów sprawdzianu krzyżowego.

Większość zbiorów z tabeli 5.2 posiada takie wartości atrybutu decyzyjnego, co do których łatwo jest stwierdzić, że jedna z nich jest bardziej pożądana od drugiej, np. w zbiorze *car-reduced* naturalnym wydaje się, że klasa *acc* jest bardziej pożądana od *unacc*. W takiej sytuacji wybór klasy źródłowej i docelowej odzwierciedla ten fakt. W pozostałych przypadkach wyboru dokonano arbitralnie (np. zbiory *vote*, *iris-reduced*). Dwa zbiory, *car* i *iris* zostały sztucznie ograniczone tak, aby zawierały przykłady tylko dwóch klas. Są to zbiory powszechnie używane w pracach badawczych, dlatego autor zdecydował się na użycie ich również w niniejszej pracy, ale by móc potraktować je na równi z pozostałymi zbiorami konieczna była ich modyfikacja wyjaśniona w opisie tabeli 5.2.

Nazwa zbioru	Liczba przykładów	Liczba atrybutów	Atrybuty nominalne	Atrybuty numeryczne	Klasa źródłowa	Klasa docelowa	XGB zb. treningowy	XGB zb. testowy
car-reduced	1594	6	6	0	unacc	acc	1.000	0.994
credit-a	690	15	9	6	bad	good	0.999	0.862
credit-g	1000	20	13	7	bad	good	0.999	0.672
diabetes-c	768	8	0	8	positive	negative	1.000	0.702
echocardiogram	131	11	2	9	dead	alive	0.993	0.864
heart-c	303	13	7	6	absent	present	1.000	0.816
heart-statlog	270	13	0	13	present	absent	1.000	0.815
hepatitis	155	19	13	6	die	live	0.990	0.696
horse-colic	368	22	15	7	not-surgical	surgical	0.996	0.842
hungarian-heart-disease	294	13	7	6	absent	present	0.999	0.751
iris-reduced	100	4	0	4	setosa	versicolor	1.000	1.000
monk1_train	124	6	6	0	0	1	0.982	0.858
mushroom	8124	22	22	0	poisonous	edible	1.000	1.000
tic-tac-toe	958	9	9	0	o-wins	x-wins	0.998	0.969
titanic	2201	3	3	0	deceased	survived	0.684	0.682
vote	435	16	16	0	republican	democrat	0.995	0.948

**Tabela 5.2:** Zbiory danych wykorzystane w eksperymentach w zadaniu klasyfikacji. Zbiór *car-reduced* powstał poprzez ograniczenie oryginalnego zbioru *car* do dwóch klas. Zbiór *iris-reduced* powstał w wyniku usunięcia z niego przykładów klasy *Iris-virginica*.

Zbiory *car-reduced* i *iris-reduced* opisują odpowiednio atrakcyjność modeli samochodów i budowę kwiatów irysa. Zbiór *credit-a* opisuje wnioski o przyznanie karty kredytowej, a *credit-g* klasyfikuje klientów banku jako dobrych lub złych kredytobiorców. Zbiór *diabetes-c* zawiera dane o pacjentach i zdiagnozowaniu cukrzycy, *echocardiogram* opisuje kilka cech związanych z sercem pacjentów po przebytych zawałach serca, a atrybutem decyzyjnym jest informacja czy pacjent przeżył co najmniej rok po zawałach. Zmienną decyzyjną w zbiorach *heart-c*, *heart-statlog* i *hungarian-heart-disease* jest informacja o tym czy pacjent choruje na jedną z chorób serca czy też nie; *hepatitis* opisuje przypadki zapalenia wątroby. *Horse-colic* zawiera dane koni leczonych na końską kolkę, zadaniem klasyfikacyjnym jest określenie, czy potrzebne jest leczenie chirurgiczne czy też nie. Zbiór *monk1\_train* był omawiany w rozdziale 3.1, *mushroom* opisuje cechy grzybów, a decyzją jest czy dany grzyb jest jadalny czy trujący. W zbiorze *tic-tac-toe* każdy wiersz opisuje konfigurację planszy do gry w kółko i krzyżyk przed wykonaniem ostatniego ruchu przez gracza, który stawia krzyżyki. Zmienną decyzyjną jest informacja o tym, czy zwyciężą krzyżyki, czy kółka. W zbiorze *titanic*, każdy wiersz reprezentuje pasażera feralnego rejsu Titanica. Zmienna decyzyjna określa, czy pasażer przeżył katastrofę. Zbiór *vote* opisuje wyniki głosowania przez członków parlamentu USA w 16 głosowaniach, oraz przypisuje każdego głosującego do partii demokratycznej lub republikańskiej.

### 5.3.2 Indukcja reguł „w przód”

#### Charakterystyka zbiorów reguł akcji

W tabeli 5.3 zebrano cechy zbiorów reguł akcji uzyskanych dla wybranych miar jakości i wariantu  $F$  indukcji. Strategie wyboru przedziałów w budowanych akcjach ustalono na *all-ranges*. Przedstawione wartości to średnie odpowiednich wyników dla wszystkich zbiorów z tabeli 5.2.

Analiza tej tabeli pokazuje, że najliczniejsze zbiory reguł indukowane są, gdy do nadzorowania procesu wzrostu i przycinania reguły wykorzystywane są miary C2 i WeightedLaplace. Podobnie, średnio najwięcej akcji elementarnych jest zawartych w przesłankach reguł dla miar C2 i WeightedLaplace. Wzorzec ten powtarza się przy analizie wartości precyzji i pokrycia części źródłowej i docelowej reguł. Tak uproszczona analiza przedstawionej tabeli pozwala stwierdzić, że zbiory reguł akcji zawierające najwięcej wiedzy o możliwych zmianach (wysoka liczba akcji elementarnych), przy

jednocześnie wysokiej precyzji i pokryciu są uzyskiwane z zastosowaniem wariantu *rules-C2-F*.

	C2	Correlation	Information Gain	RSS	Weighted Laplace
liczba reguł	11.86	5.62	6.00	3.87	14.19
liczba warunków elementarnych	3.62	3.46	3.45	3.23	3.32
liczba akcji elementarnych	1.81	1.35	1.43	1.17	1.85
precyzja części źródłowej	0.93	0.83	0.84	0.78	0.94
precyzja części docelowej	0.93	0.85	0.86	0.83	0.94
pokrycie części źródłowej	0.43	0.62	0.62	0.68	0.31
pokrycie części docelowej	0.41	0.63	0.60	0.66	0.29

**Tabela 5.3:** Wybrane charakterystyki zbiorów reguł akcji wariantu *F*. Wartości uśrednione z 16 zbiorów danych i 10 przebiegów sprawdzianu krzyżowego.

Tabela 5.4 zawiera charakterystyki zbiorów reguł akcji uzyskanych dla każdego z testowych zbiorów danych przy użyciu wariantu *rules-C2-F-all-ranges*. Liczba uzyskanych reguł waha się od 1 (zbiór *iris-reduced*) do 43 (*credit-g*), średnia liczba akcji na regułę wynosi od 1 (ponownie *iris-reduced*) do 2.68 dla zbioru *tic-tac-toe*.

Szczególną uwagę zwracają na siebie wartości podane dla zbioru *iris-reduced*, dla którego algorytm znajduje jedną regułę dokładną o wysokim pokryciu zarówno w klasie docelowej jak i źródłowej. Reguła ta w niemal wszystkich przebiegach sprawdzianu krzyżowego przybiera następującą postać:

**IF** (petallength,  $(-\infty, 2.45) \rightarrow < 3.15, \infty$ ) **THEN** (class, Iris-setosa  $\rightarrow$  Iris-versicolor)

Dla porównania, klasyczny algorytm indukcji reguł klasyfikacyjnych, nadzorowany miarą C2, na tym samym zbiorze treningowym generuje następujący zbiór reguł:

$r1$  : **IF** (petallength,  $(-\infty, 2.45)$ ) **THEN** (class, Iris-setosa)

$r2$  : **IF** (petallength,  $(2.45, \infty)$ ) **THEN** (class, Iris-versicolor)

Obydwa algorytmy wyznaczają atrybut *petallength* jako najlepiej separujący klasy, natomiast wartość cięcia dobierana jest w inny sposób. Jest to naturalnie konsekwencją zasady działania algorytmu indukcji reguł akcji. W tym przypadku jednak koszt wyboru innego cięcia jest niewielki, ponieważ docelowa część reguły akcji nie pokrywa tylko jednego z czterdziestu przykładów klasy *Iris-versicolor* dostępnych w zbiorze treningowym.

zbiór	#reguł	#warunków elementarnych	#akcji	pokrycie źródło	pokrycie cel	precyzja źródło	precyzja cel
car-reduced	9.5(0.71)	2(0.1)	1.66(0.21)	0.17(0.02)	0.19(0.02)	0.97(0)	0.42(0.01)
credit-a	11.8(0.92)	5.36(0.19)	1.72(0.34)	0.4(0.05)	0.46(0.08)	0.95(0.01)	0.94(0.01)
credit-g	43(0.82)	4.61(0.28)	1.7(0.15)	0.06(0.01)	0.04(0.01)	0.88(0.01)	0.99(0.01)
diabetes-c	33.6(1.71)	5.02(0.35)	2.57(0.28)	0.14(0.04)	0.2(0.03)	0.94(0.02)	0.99(0)
echocardiogram	4.6(0.97)	3.02(0.24)	1.15(0.14)	0.72(0.06)	0.78(0.04)	0.99(0.01)	0.97(0.01)
heart-c	14.4(1.84)	4.76(0.37)	2.66(0.27)	0.37(0.05)	0.27(0.05)	0.97(0.01)	0.97(0.02)
heart-statlog	11.2(1.62)	3.89(0.37)	2.63(0.15)	0.34(0.04)	0.4(0.04)	0.96(0.01)	0.94(0.02)
hepatitis	3.8(0.79)	4.62(0.81)	1.71(0.29)	0.4(0.05)	0.44(0.1)	0.9(0.07)	0.99(0.01)
horse-colic	8.8(1.14)	5.19(0.25)	2.26(0.4)	0.51(0.08)	0.48(0.12)	0.81(0.05)	0.93(0.01)
hungarian-heart-disease	9.2(1.03)	4.72(0.4)	2.38(0.15)	0.47(0.04)	0.34(0.08)	0.95(0.01)	0.94(0.02)
iris-reduced	1(0)	1(0)	1(0)	1(0)	0.98(0)	1(0)	1(0)
monk1_train	8(0.67)	2.25(0.21)	1.14(0.15)	0.21(0.03)	0.42(0.04)	0.85(0.03)	0.96(0.04)
mushroom	7.8(0.42)	2.7(0.1)	1.14(0.05)	0.54(0.01)	0.37(0.02)	0.98(0)	0.98(0)
tic-tac-toe	16.7(1.25)	3.46(0.18)	2.68(0.2)	0.12(0.01)	0.08(0.02)	0.9(0.02)	0.97(0.01)
titanic	4.1(0.32)	2.51(0.03)	1.55(0.12)	0.52(0.07)	0.15(0.01)	0.82(0.01)	0.98(0.01)
vote	2.2(0.42)	2.83(0.3)	1(0)	0.83(0.01)	0.95(0.01)	0.97(0.01)	0.98(0)

**Tabela 5.4:** Charakterystyki zbiorów reguł akcji uzyskanych przy użyciu wariantu  $C2-F-all-ranges$  dla każdego z analizowanych zbiorów. Dane uśrednione z 10 przebiegów. W nawiasie podano odchylenie standardowe.

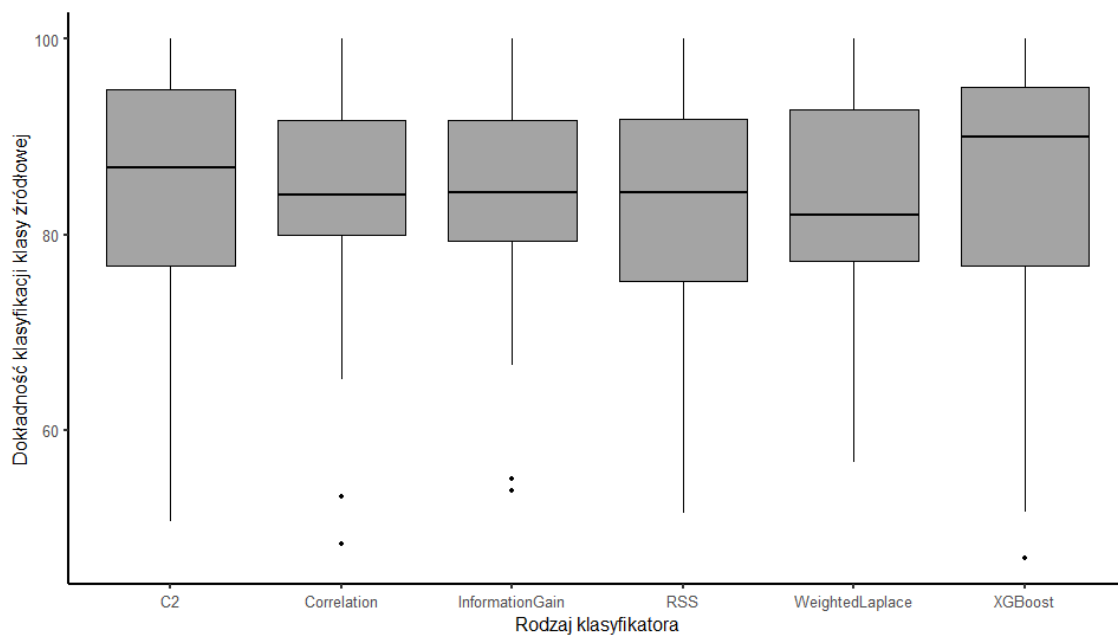
### Skuteczność transformacji

Z punktu widzenia praktycznego informacja o skuteczności modelu akcji wydaje się być cenniejsza niż jego możliwości opisowe. Dalsza część rozdziału prezentuje analizę reguł akcji „w przód” oraz algorytmu rekomendacyjnego pod kątem ich zdolności do skutecznej transformacji nieznanymi wcześniej przykładów.

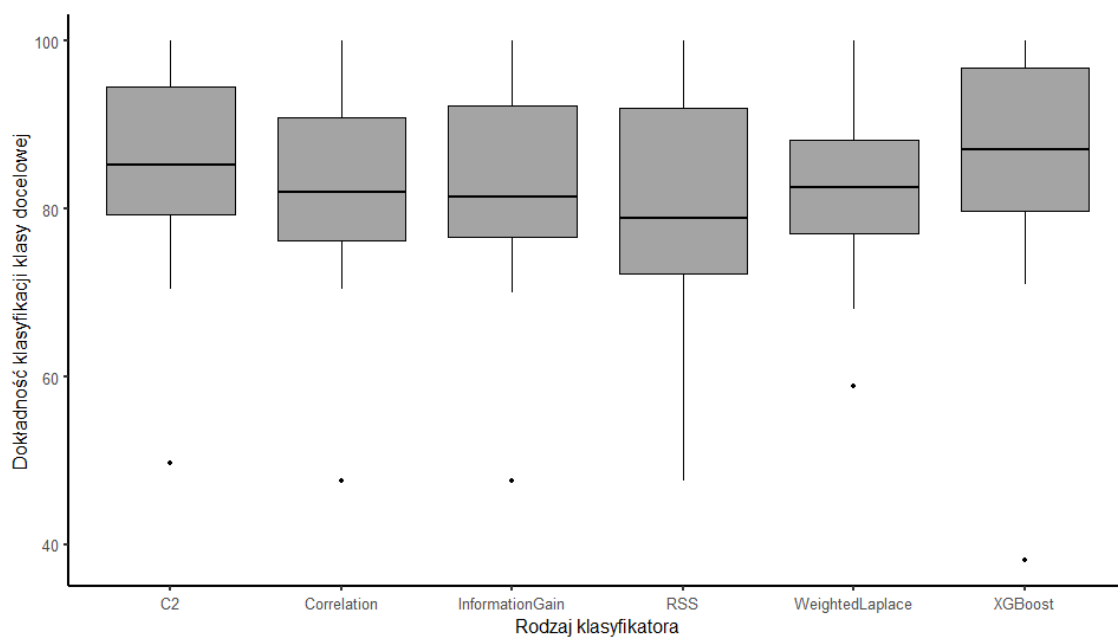
Średnią dokładność klasyfikacji klasy źródłowej i docelowej w testowych zbiorach przedstawiono w postaci dwóch wykresów pudełkowych, odpowiednio na rysunku 5.2 dla klasy źródłowej i na rysunku 5.3 dla klasy docelowej. Na osi X oznaczono algorytm XGBoost oraz 5 nazw miar jakości klasyfikacji oznaczających odpowiedni klasyfikator regułowy (do którego treningu wykorzystana została dana miara) – są to te same miary, które użyte zostały do trenowania modeli akcyjnych. Na osi Y znajduje się dokładność klasyfikacji wyrażona w procentach. Wykresy wyznaczono na podstawie danych pochodzących z 10-krotnego sprawdzianu krzyżowego przeprowadzonego dla 16 testowych zbiorów. Analiza zamieszczonych wykresów pozwala stwierdzić, że klasyfikator XGBoost średnio uzyskuje najwyższą dokładność zarówno dla klasy źródłowej jak i docelowej. Dokładność klasyfikacji przy użyciu algorytmu XGBoost na zbiorach testowych i treningowych wykorzystywanych w eksperymentach podano w tabeli 5.2.

W tabeli 5.5 przedstawiono skuteczność transformacji, tj. wartość metryki określonej wzorem 4.5 dla algorytmu rekomendacji i metody transformacji przykładów przy użyciu zbioru reguł w 5 wariantach parametryzowanych miarami jakości. Dla łatwiejszego porównania, wyniki są zgrupowane w ramach zbioru testowego, a wartości lepsze w danej grupie zostały wyróżnione. Algorytm rekomendacji okazuje się być skuteczniejszy częściej niż strategia transformacji najlepszą regułą, jednak przewaga ta nie jest dominująca.

## 5 Eksperymenty



**Rysunek 5.2:** Wykres pudełkowy dokładności klasyfikacji klasy źródłowej w testowych zbiorach danych przez klasyfikatory regułowe nadzorowane jedną z pięciu miar i algorytm XGBoost.



**Rysunek 5.3:** Wykres pudełkowy dokładności klasyfikacji klasy docelowej w testowych zbiorach danych przez klasyfikatory regułowe nadzorowane jedną z pięciu miar i algorytm XGBoost.



## 5 Eksperymenty

zbiór	algorytm	C2	Correlation	Information Gain	RSS	Weighted Laplace
car-reduced	rekomendacje	<b>0.994</b>	<b>0.714</b>	<b>0.716</b>	0.455	<b>0.994</b>
car-reduced	najlepsza reguła	0.649	0.553	0.603	<b>0.590</b>	0.649
credit-a	rekomendacje	0.708	<b>0.561</b>	<b>0.573</b>	<b>0.558</b>	0.594
credit-a	najlepsza reguła	<b>0.897</b>	0.557	0.562	0.539	<b>0.853</b>
credit-g	rekomendacje	0.648	<b>0.932</b>	<b>0.933</b>	<b>0.917</b>	0.683
credit-g	najlepsza reguła	<b>0.850</b>	0.894	0.906	0.849	<b>0.887</b>
diabetes-c	rekomendacje	0.750	0.844	0.913	0.785	0.776
diabetes-c	najlepsza reguła	<b>0.986</b>	<b>0.925</b>	<b>0.956</b>	<b>0.848</b>	<b>0.940</b>
echocardiogram	rekomendacje	<b>0.961</b>	0.956	<b>0.956</b>	<b>0.956</b>	<b>0.961</b>
echocardiogram	najlepsza reguła	0.957	<b>0.959</b>	0.953	0.955	0.958
heart-c	rekomendacje	0.436	0.427	0.506	<b>0.467</b>	0.524
heart-c	najlepsza reguła	<b>0.847</b>	<b>0.529</b>	<b>0.570</b>	0.379	<b>0.603</b>
heart-statlog	rekomendacje	0.375	0.629	<b>0.700</b>	0.550	0.496
heart-statlog	najlepsza reguła	<b>0.772</b>	<b>0.671</b>	0.689	<b>0.632</b>	<b>0.637</b>
hepatitis	rekomendacje	0.733	<b>0.667</b>	<b>0.800</b>	0.800	0.717
hepatitis	najlepsza reguła	<b>0.848</b>	0.653	0.747	<b>0.868</b>	<b>0.856</b>
horse-colic	rekomendacje	<b>0.944</b>	0.904	0.904	0.922	<b>0.952</b>
horse-colic	najlepsza reguła	0.942	<b>0.911</b>	<b>0.910</b>	<b>0.941</b>	0.930
hungarian-heart-disease	rekomendacje	<b>0.689</b>	<b>0.292</b>	<b>0.426</b>	<b>0.295</b>	0.368
hungarian-heart-disease	najlepsza reguła	0.545	0.287	0.393	0.275	<b>0.475</b>
iris-reduced	rekomendacje	1.000	1.000	1.000	1.000	1.000
iris-reduced	najlepsza reguła	1.000	1.000	1.000	1.000	1.000
monk1_train	rekomendacje	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>
monk1_train	najlepsza reguła	0.916	0.937	0.930	0.860	0.886
mushroom	rekomendacje	<b>0.976</b>	<b>0.944</b>	<b>0.944</b>	<b>0.944</b>	<b>0.080</b>
mushroom	najlepsza reguła	0.613	0.604	0.584	0.629	0.073
tic-tac-toe	rekomendacje	0.947	0.798	0.806	0.759	0.936
tic-tac-toe	najlepsza reguła	<b>0.994</b>	<b>0.976</b>	<b>0.988</b>	<b>0.836</b>	<b>0.994</b>
titanic	rekomendacje	1.000	<b>0.718</b>	<b>0.718</b>	<b>0.718</b>	1.000
titanic	najlepsza reguła	1.000	0.651	0.651	0.650	1.000
vote	rekomendacje	0.988	0.982	0.988	0.988	<b>0.985</b>
vote	najlepsza reguła	0.988	<b>0.988</b>	0.988	0.988	0.903
średnia skuteczność	rekomendacje	0.822	0.773	0.805	0.757	0.754
średnia skuteczność	najlepsza reguła	0.863	0.756	0.777	0.740	0.790

**Tabela 5.5:** Skuteczność transformacji przykładów ze zbioru testowego określona metryką  $m_2$ . W ramach każdego ze zbiorów pogrubiono wynik lepszy dla danej miary. W przypadku takich samych wyników dla danego przypadku, żadna z wartości nie została wyróżniona. Ostatnie dwa wiersza zawierają średnią skuteczność uzyskaną na wszystkich zbiorach testowych.

Metryka  $m_2$  (wzór 4.5) nie uwzględnia niestety liczby przykładów, dla których model akcyjny wygenerował akcje. Opierając się tylko o tę metrykę, moglibyśmy uznać, że model akcyjny który generuje skuteczne akcje, ale bardzo rzadko, jest lepszy niż taki, które popełnia niewiele błędów, ale proponuje akcje dla każdego przykładu. Dlatego porównując ze sobą warianty algorytmów, należy wziąć również pod uwagę to, jak często dana metoda w ogóle podaje rekomendacje dla przykładu. Dane takie dla wszystkich zbiorów testowych i wariantów  $F$  algorytmu rekomendacji i transformacji przykładów najlepszą regułą zebrano w tabeli 5.6. Jej analiza pokazuje, że algorytm indukcji rekomendacji przewyższa w tym względzie strategię transformacji przykładów przy użyciu najlepszej reguły w zbiorze. Dla niemal każdego ze zbiorów testów algorytm rekomendacji był w stanie pokryć wszystkie przykłady testowe, podczas gdy metoda regułowa, choć uzyskuje na ogół wysoki wynik pokrycia, to nigdy nie wynosi ono 100%.

Idąc dalej, poszczególne warianty metody można porównywać pod kątem precyzji części źródłowej i docelowej rekomendacji, ich pokrycia w zbiorze treningowym, tego, jak wiele akcji elementarnych jest niezbędnych do skutecznego przeniesienia przykładu czy też jak szerokie zakresy wartości są używane w tychże akcjach elementarnych. Kryteria jakości metody transformacji przykładów można mnożyć, a i tak trudnym, czy wręcz niemożliwym, będzie podanie kryterium, które pozwoli jednoznacznie określić, która metoda będzie najlepszym wyborem w każdym potencjalnym zastosowaniu. Intencją autora jest przedstawienie idei i kluczowych wskaźników które należy wziąć pod uwagę przy wyborze algorytmu, co jak wykazało studium literatury, nie było tematem do tej pory szeroko omawianym w dziedzinie reguł akcji. W dalszym ciągu rozdziału podobna do przedstawionej w tym rozdziale analiza zostanie przeprowadzona dla pozostałych wariantów metody oraz dla regresyjnych reguł akcji. Metody przedstawione w tej pracy zostaną ze sobą zestawione pod kątem skuteczności transformacji, co powinno być wystarczające, by pokierować i pomóc potencjalnemu czytelnikowi w wyborze metody, która najlepiej sprawdzi się w zadaniu, przed którym stanie.

## 5 Eksperymenty

zbiór	algorytm	C2	Correlation	Information Gain	RSS	Weighted Laplace
car-reduced	rekomendacje	100.0	100.0	100.0	100.0	100.0
car-reduced	najlepsza reguła	95.7	95.5	95.5	95.5	95.7
credit-a	rekomendacje	100.0	100.0	100.0	99.6	100.0
credit-a	najlepsza reguła	92.3	96.1	95.7	95.6	88.4
credit-g	rekomendacje	100.0	100.0	100.0	100.0	100.0
credit-g	najlepsza reguła	72.7	90.7	92.2	93.5	64.7
diabetes-c	rekomendacje	100.0	100.0	100.0	100.0	100.0
diabetes-c	najlepsza reguła	78.7	93.1	94.6	95.6	71.9
echocardiogram	rekomendacje	100.0	100.0	100.0	100.0	100.0
echocardiogram	najlepsza reguła	93.3	92.8	93.3	96.7	90.0
heart-c	rekomendacje	100.0	100.0	100.0	100.0	100.0
heart-c	najlepsza reguła	88.5	93.0	91.5	92.4	85.8
heart-statlog	rekomendacje	100.0	100.0	100.0	100.0	100.0
heart-statlog	najlepsza reguła	82.9	85.4	88.3	90.4	79.2
hepatitis	rekomendacje	100.0	100.0	100.0	100.0	100.0
hepatitis	najlepsza reguła	95.0	73.3	70.0	73.3	88.3
horse-colic	rekomendacje	100.0	100.0	100.0	100.0	100.0
horse-colic	najlepsza reguła	89.6	90.7	90.4	92.2	84.1
hungarian-heart-disease	rekomendacje	100.0	100.0	100.0	100.0	100.0
hungarian-heart-disease	najlepsza reguła	88.9	95.8	94.5	94.5	87.6
iris-reduced	rekomendacje	100.0	100.0	100.0	100.0	100.0
iris-reduced	najlepsza reguła	100.0	100.0	100.0	100.0	100.0
monk1_train	rekomendacje	100.0	100.0	100.0	100.0	100.0
monk1_train	najlepsza reguła	83.3	84.2	82.5	85.0	85.8
mushroom	rekomendacje	100.0	100.0	100.0	100.0	100.0
mushroom	najlepsza reguła	95.6	95.3	95.3	95.9	95.9
tic-tac-toe	rekomendacje	100.0	100.0	100.0	100.0	100.0
tic-tac-toe	najlepsza reguła	96.7	96.1	96.5	98.9	96.8
titanic	rekomendacje	100.0	100.0	100.0	100.0	100.0
titanic	najlepsza reguła	97.3	97.5	97.5	97.7	97.3
vote	rekomendacje	99.1	99.4	96.5	97.1	99.4
vote	najlepsza reguła	95.6	97.1	97.9	97.6	93.5

**Tabela 5.6:** Liczba przykładów ze zbioru testowego, dla których wygenerowano rekomendacje, wyrażona w procentach. Dane uśrednione z 10 przebiegów sprawdzianu krzyżowego.

### 5.3.3 Indukcja reguł „wstecz”

#### Charakterystyka zbiorów reguł akcji

W tabeli 5.7 zebrano cechy zbiorów reguł akcji uzyskanych dla wybranych miar jakości i wariantu  $B$  metody indukcji reguł akcji. Strategie wyboru przedziałów w budowanych akcjach ustalono na *all-ranges*. Przedstawione wartości to średnie odpowiednich wyników dla wszystkich zbiorów z tabeli 5.2.

Analizując charakterystyki zbiorów reguł pod kątem wariantu najlepiej opisującego wiedzę dotyczącą przejść przykładów między klasami źródłową i docelową, podobnie jak w przypadku wariantu  $F$  można dostrzec zależność, że najwyższą średnią liczbę akcji elementarnych uzyskuje się stosując miary C2 i WeightedLaplace do nadzorowania algorytmu indukcji reguł. Te same warianty metody pozwalają uzyskać reguły o najwyższej wartości precyzji zarówno części źródłowej jak i docelowej reguły. Reguły akcji o najwyższej wartości pokrycia przez część źródłową i docelową uzyskuje się stosując miarę RSS. Miary C2 i WeightedLaplace dają zbiory reguł liczniejsze, ale charakteryzujące się niższą średnią wartością pokrycia niż RSS.

	C2	Correlation	Information Gain	RSS	Weighted Laplace
liczba reguł	15.43	5.21	6.38	4.38	19.82
liczba warunków elementarnych	3.52	3.35	3.37	3.23	3.25
liczba akcji elementarnych	2.89	2.57	2.65	2.37	2.71
precyzja części źródłowej	0.92	0.84	0.84	0.79	0.91
precyzja części docelowej	0.93	0.85	0.86	0.81	0.95
pokrycie części źródłowej	0.44	0.64	0.64	0.71	0.34
pokrycie części docelowej	0.43	0.61	0.60	0.66	0.32

**Tabela 5.7:** Wybrane charakterystyki zbiorów reguł akcji wariantu  $B$ . Wartości uśrednione z 16 zbiorów danych i 10 przebiegów.

Podobnie jak w podrozdziale 5.3.2 dotyczącej wariantu  $F$ , w dalszej części przedstawiono bardziej szczegółowe informacje o wynikach uzyskanych przez metodę *rules-C2-B-all-ranges*. W tabeli 5.8 podano podstawowe charakterystyki zbiorów reguł akcji uzyskanych na każdym z testowych zbiorów danych. Podobnie jak w przypadku wariantu *C2-F-all-ranges*, dla zbioru *iris-reduced* wyindukowana została tylko jedna reguła o następującej postaci:

**IF** (petallength,  $(-\infty, 1.80) \rightarrow < 2.45, \infty$ ) **THEN** (class, Iris-setosa  $\rightarrow$  Iris-versicolor)

Reguła ta jest bardzo zbliżona do jej odpowiednika odkrytego przez wariant  $F$ , różni się tylko punktem cięcia na atrybucie *petallength*. Innym ciekawym przypadkiem jest zbiór *monk1\_train*. Dla przypomnienia, w zbiorze tym przykłady są przypisane do klasy docelowej (tj. klasy „1”) wówczas, gdy zachodzi jedna z dwóch zależności:  $attr5 = 1$  lub  $attr1 = attr2$ . Druga z wymienionych zależności, w swojej ogólnej postaci, jest niemożliwa do odkrycia przez algorytm indukcji reguł, ponieważ w trakcie odkrywania reguł nie są badane zależności funkcyjne między atrybutami. Pomimo tego, w uzyskanych zbiorach reguł można spotkać reguły, które w swojej przesłance poprawnie opisują przypadki szczególne tej zależności, np.:

**IF** ( $attr2, 3 \rightarrow 3$ )  $\wedge$  ( $attr1, 1 \rightarrow 3$ ) **THEN** ( $class, 0 \rightarrow 1$ )

Przesłanka tej reguły dobrze ilustruje sens indukowania akcji podtrzymujących. W tym przypadku źródłowa część pierwszego warunku elementarnego wybiera podzbiór przykładów zbioru treningowego o ustalonej wartości atrybutu *attr2*, a jego docelowa część nakazuje utrzymanie tej wartości, co pozwala - po dołączeniu drugiej akcji elementarnej - na poprawne odwzorowanie szczególnego przypadku reguły kwalifikującej przykłady do klasy docelowej.

zbiór	#reguł	#warunków elementarnych	#akcji	pokrycie źródło	pokrycie cel	precyzja źródło	precyzja cel
car-reduced	35.6(1.26)	3.67(0.05)	3.67(0.05)	0.48(0.01)	0.04(0)	1(0)	0.93(0.01)
credit-a	10.5(2.37)	4.52(0.53)	3.88(0.45)	0.67(0.07)	0.52(0.08)	0.96(0.01)	0.88(0.02)
credit-g	61.4(2.46)	4.11(0.1)	3.77(0.14)	0.05(0.02)	0.1(0.01)	0.81(0.03)	0.98(0)
diabetes-c	46(2.91)	4.57(0.2)	4.4(0.18)	0.1(0.03)	0.24(0.01)	0.97(0.01)	0.97(0)
echocardiogram	2.1(0.32)	3.07(0.39)	2.78(0.31)	0.67(0.04)	0.78(0.07)	0.98(0.01)	0.88(0.03)
heart-c	11.8(1.75)	3.76(0.13)	3.43(0.21)	0.27(0.04)	0.35(0.04)	0.95(0.01)	0.95(0.01)
heart-statlog	10.1(1.2)	4.92(0.36)	4.67(0.33)	0.33(0.03)	0.44(0.04)	0.98(0.01)	0.98(0.01)
hepatitis	7.5(1.27)	4.09(0.39)	3.36(0.26)	0.35(0.05)	0.55(0.06)	0.75(0.06)	0.98(0.01)
horse-colic	10.4(1.84)	4.89(0.26)	3.56(0.26)	0.51(0.06)	0.42(0.04)	0.88(0.04)	0.93(0.01)
hungarian-heart-disease	9.5(0.97)	5.05(0.35)	4.35(0.26)	0.38(0.07)	0.46(0.05)	0.98(0.02)	0.86(0.03)
iris-reduced	1(0)	1(0)	1(0)	0.95(0.02)	1(0)	1(0)	1(0)
monk1_train	3.9(0.32)	1.74(0.21)	1.33(0.17)	0.28(0.03)	0.33(0.07)	0.78(0.06)	0.93(0.09)
mushroom	4.4(0.52)	1.31(0.34)	1.1(0.11)	0.59(0.01)	0.33(0.06)	0.94(0.02)	0.99(0)
tic-tac-toe	22.2(4.92)	3.54(0.14)	2.75(0.17)	0.11(0.01)	0.09(0.01)	0.88(0.01)	0.99(0.01)
titanic	8.9(0.99)	2.85(0.05)	1.13(0.11)	0.46(0.13)	0.29(0.03)	0.85(0.02)	0.7(0.03)
vote	1.6(0.52)	3.25(0.68)	1(0)	0.83(0.03)	0.93(0.02)	0.97(0.01)	0.96(0.02)

**Tabela 5.8:** Charakterystyki zbiorów reguł akcji uzyskanych przy użyciu wariantu *C2-B-all-ranges* dla każdego z analizowanych zbiorów. Dane uśrednione z 10 przebiegów. W nawiasie podano odchylenie standardowe.

### Skuteczność transformacji

Skuteczność transformacji określona wartością miary 4.5 dla algorytmu transformacji przykładów przy użyciu najlepszej reguły i algorytmu rekomendacji w wariancie *B* została zebrana w tabeli 5.9. Wyniki przedstawiono dla 5 różnych miar jakości nadzorujących proces budowania reguł i rekomendacji. Dane zostały dodatkowo pogrupowane tak, aby skuteczność każdego z algorytmów na danym zbiorze testowym była łatwa do porównania. Podobnie jak w przypadku wariantu *F*, choć algorytm rekomendacji częściej uzyskuje wyższy wynik, to żadna z metod znacząco nie przeważa nad drugą. Do dalszej analizy skuteczności niezbędna jest analiza tabeli 5.10, informująca o tym, jak często dany model akcyjny był w stanie wygenerować rekomendację dla przedstawionego przykładu testowego.

W wyborze najbardziej pożądanej metody dla danego zbioru danych pomocne może być zwizualizowanie wyników poszczególnych wariantów w postaci wykresu. Na osi OX takiego wykresu można odłożyć procent przykładów testowych, dla których wytworzono akcje (wartość można odczytać z tabeli 5.10), zaś na osi OY należy umieścić wartości odczytane z tabeli 5.9. Punktami danych będą poszczególne warianty algorytmów, które uzyskały odpowiadające wyniki. Na takim wykresie najbardziej interesujące będą metody, które znajdują się w prawym górnym rogu wykresu. Dla przykładu, na rysunku 5.4 przedstawiono wykres zbudowany według tego opisu dla wariantu *B* i zbioru *car-reduced*. Łatwo z niego odczytać, że metody charakteryzujące się maksymalnym pokryciem i wysoką skutecznością na tym zbiorze to dwa warianty algorytmu rekomendacji: nadzorowanego miarą C2 i WeightedLaplace. Zaprezentowany wykres przedstawia pokrycie zbioru testowego w liczbie przykładów, dla których wygenerowano akcje. Pokrycie przedstawiono na wykresie w zakresie wartości 0-100, jako procent przykładów pokrytych. Model akcji oparty o algorytm rekomendacji charakteryzuje się większym pokryciem, dzięki swojej konstrukcji, która oprócz wykorzystywania zakresów odkrytych przez reguły, na podstawie których budowane są rekomendacji, uwzględnia także dodatkowe zakresy pozwalające pokryć skrajne wartości atrybutów.

## 5 Eksperymenty

zbiór	algorytm	C2	Correlation	Information Gain	RSS	Weighted Laplace
car-reduced	rekomendacje	0.994	0.714	0.716	<b>0.455</b>	0.994
car-reduced	najlepsza reguła	<b>1.000</b>	<b>0.782</b>	<b>0.779</b>	0.406	<b>1.000</b>
credit-a	rekomendacje	0.717	<b>0.544</b>	<b>0.549</b>	<b>0.546</b>	0.208
credit-a	najlepsza reguła	<b>0.910</b>	0.454	0.451	0.471	<b>0.838</b>
credit-g	rekomendacje	0.652	<b>0.932</b>	<b>0.930</b>	<b>0.925</b>	0.697
credit-g	najlepsza reguła	<b>0.883</b>	0.903	0.912	0.901	<b>0.910</b>
diabetes-c	rekomendacje	0.689	0.843	0.856	0.854	0.724
diabetes-c	najlepsza reguła	<b>0.981</b>	<b>0.937</b>	<b>0.990</b>	<b>0.951</b>	<b>0.895</b>
echocardiogram	rekomendacje	<b>0.978</b>	0.978	0.978	0.944	1.000
echocardiogram	najlepsza reguła	0.977	<b>1.000</b>	<b>1.000</b>	0.994	1.000
heart-c	rekomendacje	0.430	<b>0.612</b>	<b>0.606</b>	<b>0.385</b>	0.527
heart-c	najlepsza reguła	<b>0.693</b>	0.440	0.454	0.355	<b>0.688</b>
heart-statlog	rekomendacje	0.483	0.592	0.696	0.625	0.446
heart-statlog	najlepsza reguła	<b>0.912</b>	<b>0.852</b>	<b>0.917</b>	<b>0.867</b>	<b>0.733</b>
hepatitis	rekomendacje	0.817	<b>0.767</b>	<b>0.783</b>	<b>0.783</b>	0.667
hepatitis	najlepsza reguła	<b>0.847</b>	0.563	0.765	0.723	<b>0.943</b>
horse-colic	rekomendacje	0.922	<b>0.907</b>	<b>0.900</b>	<b>0.919</b>	<b>0.956</b>
horse-colic	najlepsza reguła	<b>0.924</b>	0.850	0.852	0.891	0.953
hungarian-heart-disease	rekomendacje	0.774	0.547	0.571	0.500	0.397
hungarian-heart-disease	najlepsza reguła	<b>0.836</b>	<b>0.618</b>	<b>0.626</b>	<b>0.586</b>	<b>0.492</b>
iris-reduced	rekomendacje	1.000	1.000	1.000	1.000	1.000
iris-reduced	najlepsza reguła	1.000	1.000	1.000	1.000	1.000
monk1_train	rekomendacje	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>
monk1_train	najlepsza reguła	0.763	0.782	0.755	0.809	0.747
mushroom	rekomendacje	<b>0.952</b>	<b>0.944</b>	<b>0.944</b>	<b>0.944</b>	0.491
mushroom	najlepsza reguła	0.698	0.635	0.635	0.635	<b>0.599</b>
tic-tac-toe	rekomendacje	0.947	0.798	0.806	0.762	0.936
tic-tac-toe	najlepsza reguła	0.947	<b>0.921</b>	<b>0.934</b>	<b>0.925</b>	<b>0.953</b>
titanic	rekomendacje	<b>1.000</b>	<b>0.718</b>	<b>0.718</b>	<b>0.718</b>	<b>1.000</b>
titanic	najlepsza reguła	0.567	0.702	0.706	0.711	0.359
vote	rekomendacje	<b>0.988</b>	0.982	0.988	0.988	<b>0.997</b>
vote	najlepsza reguła	0.921	<b>0.988</b>	0.988	0.988	0.980
średnia skuteczność	rekomendacje	0.834	0.805	0.815	0.772	0.752
średnia skuteczność	najlepsza reguła	0.866	0.777	0.798	0.763	0.818

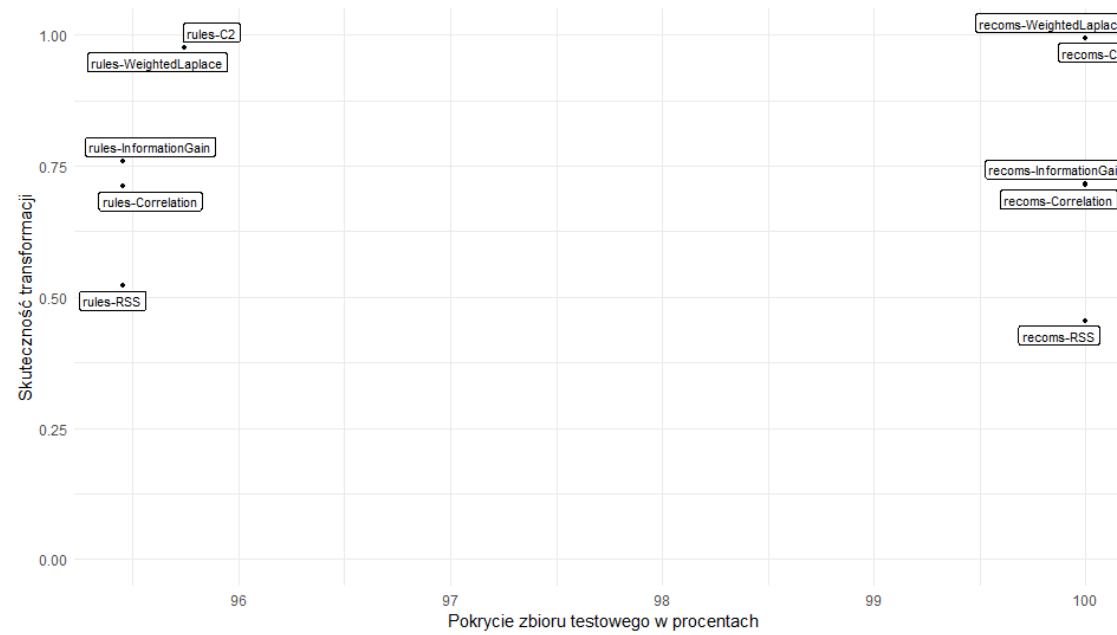
**Tabela 5.9:** Skuteczność transformacji  $m_2$  przykładów ze zbioru testowego z zastosowaniem wariantu  $B$  algorytmu indukcji reguł i rekomendacji. W ramach każdego ze zbiorów pogrubiono wynik lepszy. W przypadku remisu, żadna z wartości nie została wyróżniona. Ostatnie dwa wiersze zawierają wyniki uśrednione dla każdej miary na wszystkich zbiorach.



## 5 Eksperymenty

zbiór	algorytm	C2	Correlation	Information Gain	RSS	Weighted Laplace
car-reduced	rekomendacje	100.0	100.0	100.0	100.0	100.0
car-reduced	najlepsza reguła	73.3	69.8	69.8	69.8	73.3
credit-a	rekomendacje	100.0	100.0	99.0	99.0	100.0
credit-a	najlepsza reguła	81.7	87.8	84.2	86.2	98.6
credit-g	rekomendacje	100.0	100.0	100.0	100.0	100.0
credit-g	najlepsza reguła	67.0	64.8	65.5	87.0	57.0
diabetes-c	rekomendacje	100.0	100.0	100.0	100.0	100.0
diabetes-c	najlepsza reguła	38.7	53.5	60.4	82.8	33.1
echocardiogram	rekomendacje	100.0	100.0	100.0	100.0	100.0
echocardiogram	najlepsza reguła	74.4	98.9	98.9	92.8	62.8
heart-c	rekomendacje	100.0	100.0	100.0	100.0	100.0
heart-c	najlepsza reguła	83.9	95.2	93.3	88.8	90.3
heart-statlog	rekomendacje	100.0	100.0	100.0	100.0	100.0
heart-statlog	najlepsza reguła	72.1	75.0	76.7	84.2	76.2
hepatitis	rekomendacje	100.0	100.0	100.0	100.0	100.0
hepatitis	najlepsza reguła	66.7	68.3	88.3	85.0	83.3
horse-colic	rekomendacje	100.0	100.0	100.0	100.0	100.0
horse-colic	najlepsza reguła	81.1	82.2	84.1	88.9	77.8
hungarian-heart-disease	rekomendacje	100.0	100.0	100.0	100.0	100.0
hungarian-heart-disease	najlepsza reguła	74.5	91.6	90.8	90.0	69.5
iris-reduced	rekomendacje	100.0	100.0	100.0	100.0	100.0
iris-reduced	najlepsza reguła	95.0	95.0	95.0	95.0	95.0
monk1_train	rekomendacje	100.0	100.0	100.0	100.0	100.0
monk1_train	najlepsza reguła	60.8	66.7	66.7	80.0	60.0
mushroom	rekomendacje	100.0	100.0	100.0	100.0	100.0
mushroom	najlepsza reguła	91.2	88.8	88.8	88.8	92.9
tic-tac-toe	rekomendacje	100.0	100.0	100.0	100.0	100.0
tic-tac-toe	najlepsza reguła	59.2	77.4	77.9	94.7	59.4
titanic	rekomendacje	100.0	100.0	100.0	100.0	100.0
titanic	najlepsza reguła	89.9	90.8	90.4	89.2	78.5
vote	rekomendacje	98.8	99.1	97.1	97.6	98.8
vote	najlepsza reguła	90.0	97.4	97.9	97.6	89.1

**Tabela 5.10:** Liczba przykładów ze zbioru testowego, dla których wygenerowano rekomendacje używając wariantu  $B$ , wyrażona w procentach względem wszystkich przykładów klasy źródłowej w zbiorze testowym. Dane uśrednione z 10 przebiegów sprawdzianu krzyżowego.



**Rysunek 5.4:** Zależność między skutecznością transformacji a stopniem pokrycia zbioru testowego dla wybranych modeli akcji.

### 5.3.4 Fuzja wariantów „w przód” i „wstecz”

Indukcja reguł akcji przy użyciu metody opartej o paradygmat sekwencyjnego pokrywania, która została przedstawiona w tej pracy, jest sterowana – w zależności od wybranego wariantu – informacjami dotyczącymi jednej z klas istotnych dla rozwiązywanego problemu. Na reguły odkryte przez wariant F największy wpływ mają przykłady klasy źródłowej, w przypadku wariantu B są to przykłady klasy docelowej. Ponieważ, po pierwsze nie wszystkie przykłady ze zbioru biorą udział w decydowaniu o postaci kolejnych reguł, a po drugie nie ma żadnej gwarancji że dany zbiór danych zawiera wszystkich możliwych reprezentantów danej klasy, proces indukcji reguł akcji jest zawodny. Jednym z pomysłów mających na celu zwiększenie skuteczności reguł akcji i powstałych na ich podstawie rekomendacji jest fuzja obu zaproponowanych wariantów. Zbiory reguł wytworzone przez wariant F i B są łączone w jeden zbiór danych, który stanowi model akcyjny oparty o reguły akcji. Na podstawie takiego połączonego zbioru reguła akcji buduje się także model rekomendacji. Dzięki temu, w trakcie transformowania przykładów, oba rodzaje modeli dysponują szerszą wiedzą o charakterystykach przykładów należących do klas źródłowej i docelowej, co powinno przełożyć się na wyższą skuteczność proponowanych akcji. W tej sekcji przedstawiono wyniki eksperymentów uzyskanych przy użyciu połączonych zbiorów reguł akcji. Ten wariant w podsumowaniach oznaczany będzie jako *FB*.

Ze względu na fakt, że przy łączeniu zbiorów reguł akcji nie przeprowadza się żadnych dodatkowych operacji filtrowania ani łączenia reguł, przedstawianie analizy ilościowej finalnego zbioru reguł akcji zostanie pominięte. Poszczególne wartości dla wariantu FB takie jak średnia liczba reguł będą średnią z odpowiednich wartości uzyskanych w wariacie F i B, możliwych do odczytania w tabelach 5.3 i 5.7. Wartości dotyczące średnich precyzji czy pokrycia nie mogą zostać uzyskane w tak łatwy sposób, jednak aby nadmiernie nie zwiększać objętości pracy, ich analiza zostanie pominięta.

#### Skuteczność transformacji

W tabeli 5.11 przedstawiono dane dotyczące skuteczności transformacji przykładów przez wariant FB w postaci bliźniaczej do tabel 5.5 i 5.9.

## 5 Eksperymenty

zbiór	algorytm	C2	Correlation	Information Gain	RSS	Weighted Laplace
car-reduced	rekomendacje	<b>0.994</b>	<b>0.714</b>	0.716	0.455	<b>0.994</b>
car-reduced	najlepsza reguła	0.976	0.711	<b>0.758</b>	<b>0.521</b>	0.976
credit-a	rekomendacje	0.691	<b>0.565</b>	<b>0.566</b>	<b>0.565</b>	0.208
credit-a	najlepsza reguła	<b>0.910</b>	0.480	0.483	0.492	<b>0.839</b>
credit-g	rekomendacje	0.690	<b>0.932</b>	<b>0.932</b>	<b>0.923</b>	0.663
credit-g	najlepsza reguła	<b>0.896</b>	0.898	0.914	0.898	<b>0.934</b>
diabetes-c	rekomendacje	0.781	0.831	0.922	0.820	0.724
diabetes-c	najlepsza reguła	<b>0.987</b>	<b>0.936</b>	<b>0.982</b>	<b>0.942</b>	<b>0.935</b>
echocardiogram	rekomendacje	<b>0.972</b>	0.978	0.978	0.956	0.961
echocardiogram	najlepsza reguła	0.965	<b>1.000</b>	<b>1.000</b>	<b>0.994</b>	<b>0.972</b>
heart-c	rekomendacje	0.348	<b>0.552</b>	<b>0.609</b>	0.327	0.506
heart-c	najlepsza reguła	<b>0.729</b>	0.458	0.475	<b>0.368</b>	<b>0.703</b>
heart-statlog	rekomendacje	0.362	0.654	0.721	0.637	0.408
heart-statlog	najlepsza reguła	<b>0.922</b>	<b>0.870</b>	<b>0.929</b>	<b>0.874</b>	<b>0.754</b>
hepatitis	rekomendacje	0.750	<b>0.700</b>	0.750	<b>0.733</b>	0.683
hepatitis	najlepsza reguła	<b>0.880</b>	0.657	<b>0.788</b>	0.730	<b>0.943</b>
horse-colic	rekomendacje	0.922	0.904	0.900	0.915	0.952
horse-colic	najlepsza reguła	<b>0.939</b>	<b>0.908</b>	<b>0.909</b>	<b>0.925</b>	<b>0.963</b>
hungarian-heart-disease	rekomendacje	0.697	0.545	0.566	0.492	0.382
hungarian-heart-disease	najlepsza reguła	<b>0.822</b>	<b>0.636</b>	<b>0.637</b>	<b>0.599</b>	<b>0.583</b>
iris-reduced	rekomendacje	1.000	1.000	1.000	1.000	1.000
iris-reduced	najlepsza reguła	1.000	1.000	1.000	1.000	1.000
monk1_train	rekomendacje	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>
monk1_train	najlepsza reguła	0.920	0.944	0.909	0.863	0.882
mushroom	rekomendacje	<b>0.976</b>	<b>0.944</b>	<b>0.944</b>	<b>0.944</b>	0.491
mushroom	najlepsza reguła	0.658	0.612	0.592	0.601	<b>0.591</b>
tic-tac-toe	rekomendacje	0.947	0.798	0.806	0.762	0.936
tic-tac-toe	najlepsza reguła	<b>0.978</b>	<b>0.943</b>	<b>0.952</b>	<b>0.913</b>	<b>0.974</b>
titanic	rekomendacje	1.000	<b>0.718</b>	<b>0.718</b>	<b>0.718</b>	1.000
titanic	najlepsza reguła	1.000	0.652	0.652	0.651	1.000
vote	rekomendacje	<b>0.988</b>	0.985	0.988	0.988	0.991
vote	najlepsza reguła	0.923	<b>0.988</b>	0.988	0.988	0.991
średnia skuteczność	rekomendacje	0.820	0.801	0.820	0.765	0.744
średnia skuteczność	najlepsza reguła	0.907	0.793	0.811	0.772	0.878

**Tabela 5.11:** Skuteczność transformacji przykładów ze zbioru testowego przy użyciu wariantu FB określona metryką  $m_2$ . Wynik lepszy dla danej miary i zbioru danych został pogrubiony. W przypadku remisu nie wyróżniono żadnej z wartości. Ostatnie dwa wiersze zawierają uśrednioną skuteczność uzyskaną na wszystkich zbiorach testowych,

Uzyskana średnia skuteczność była wyższa dla strategii transformacji najlepszą regułą w 42 przypadkach, podczas gdy rekomendacje okazały się lepsze 27 razy. W porównaniu do działania reguł w wariantach F i B osobno jest to zauważalna różnica wskazująca na wyższość pierwszej z metod, tj. reguł. Potrzeba staranniejszej analizy statystycznej, by móc to stwierdzić z większą dozą pewności. Analizę taką przeprowadzono w następnej części rozdziału.

Wyniki pokrycia uzyskane przez wariant FB przedstawiono w tabeli 5.12. Jej analiza i porównanie z tabelami 5.6 oraz 5.10 pozwala stwierdzić, że choć fuzja zbiorów reguł F i B zdecydowanie poprawia wartość pokrycia metody transformacji najlepszą regułą, to nadal algorytm rekomendacji pozwala na uzyskanie sugestii akcji dla większej liczby przykładów.

## 5 Eksperymenty

zbiór	algorytm	C2	Correlation	Information Gain	RSS	Weighted Laplace
car-reduced	rekomendacje	100.0	100.0	100.0	100.0	100.0
car-reduced	najlepsza reguła	95.7	95.5	95.5	95.5	95.7
credit-a	rekomendacje	100.0	100.0	100.0	100.0	100.0
credit-a	najlepsza reguła	93.1	97.1	96.6	96.1	98.6
credit-g	rekomendacje	100.0	100.0	100.0	100.0	100.0
credit-g	najlepsza reguła	86.7	93.2	92.8	95.7	79.0
diabetes-c	rekomendacje	100.0	100.0	100.0	100.0	100.0
diabetes-c	najlepsza reguła	82.2	93.1	94.6	96.5	74.6
echocardiogram	rekomendacje	100.0	100.0	100.0	100.0	100.0
echocardiogram	najlepsza reguła	95.6	100.0	99.4	98.9	91.1
heart-c	rekomendacje	100.0	100.0	100.0	100.0	100.0
heart-c	najlepsza reguła	93.6	98.2	97.0	95.8	95.2
heart-statlog	rekomendacje	100.0	100.0	100.0	100.0	100.0
heart-statlog	najlepsza reguła	84.2	86.7	90.8	92.1	86.7
hepatitis	rekomendacje	100.0	100.0	100.0	100.0	100.0
hepatitis	najlepsza reguła	73.3	85.0	93.3	93.3	85.0
horse-colic	rekomendacje	100.0	100.0	100.0	100.0	100.0
horse-colic	najlepsza reguła	91.1	92.6	93.0	93.7	91.1
hungarian-heart-disease	rekomendacje	100.0	100.0	100.0	100.0	100.0
hungarian-heart-disease	najlepsza reguła	91.8	97.1	96.1	96.1	90.8
iris-reduced	rekomendacje	100.0	100.0	100.0	100.0	100.0
iris-reduced	najlepsza reguła	100.0	100.0	100.0	100.0	100.0
monk1_train	rekomendacje	100.0	100.0	100.0	100.0	100.0
monk1_train	najlepsza reguła	89.2	90.8	89.2	92.5	91.7
mushroom	rekomendacje	100.0	100.0	100.0	100.0	100.0
mushroom	najlepsza reguła	98.6	95.3	95.3	95.9	98.7
tic-tac-toe	rekomendacje	100.0	100.0	100.0	100.0	100.0
tic-tac-toe	najlepsza reguła	98.0	97.9	98.0	99.5	98.2
titanic	rekomendacje	100.0	100.0	100.0	100.0	100.0
titanic	najlepsza reguła	97.5	97.8	97.8	97.9	97.5
vote	rekomendacje	99.1	99.4	97.1	97.6	99.4
vote	najlepsza reguła	95.9	97.4	97.9	97.6	96.2

**Tabela 5.12:** Procent przykładów ze zbioru testowego, dla których wygenerowano rekomendacje używając wariantu *FB*. Dane uśrednione z 10 przebiegów sprawdzianu krzyżowego.

### 5.3.5 Wybór najlepszego wariantu

W poprzedzających tę część sekcjach dokonano analizy jakościowej uzyskiwanych zbiorów reguł oraz analizy skuteczności transformacji dla modeli akcyjnych zbudowanych ze zbiorów reguł akcji i algorytmu rekomendacji. Analiza ta wykonana została z rozróżnieniem na trzy warianty oraz pięć miar nadzorujących wzrost reguły. Analiza ta nie podaje niestety prostej konkluzji i nie pozwala jednoznacznie stwierdzić, że jedna z przedstawionych metod zdecydowanie góruje nad pozostałymi. Przeprowadzona analiza opierała się o proste porównywanie uśrednionych wyników uzyskanych na zbiorach testowych. Aby spróbować udzielić wiarygodnej odpowiedzi na pytanie „który wariant metody jest najskuteczniejszy w transformacji przykładów” należy posłużyć się analizą statystyczną, która pozwoli ustalić, czy uzyskane dane zawierają wystarczająco wiele dowodów na to, że jeden z algorytmów jest lepszy od innego. Procedurą pozwalającą rozstrzygnąć ten problem jest przeprowadzenie odpowiednich testów statystycznych, które pozwolą zweryfikować hipotezy dotyczące skuteczności poszczególnych algorytmów.

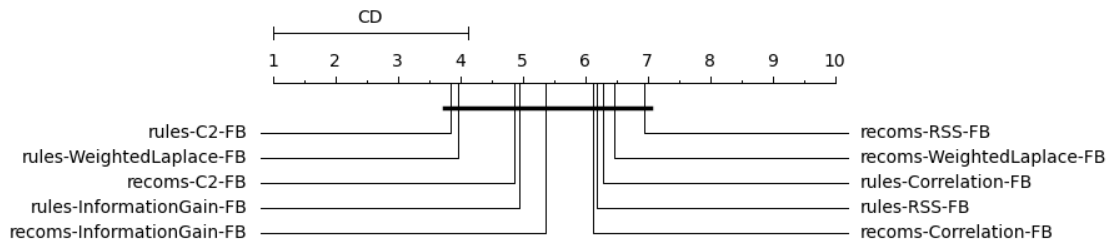
#### Analiza statystyczna uzyskanych wyników

Porównano ze sobą grupę  $2 \times 3 \times 5 = 30$  ( $\{reguły, rekomendacje\} \times \{F, B, FB\} \times \{C2, Correlation, InformationGain, RSS, WeightedLaplace\}$ ) algorytmów. Strategia wyboru przedziałów została ustalona na *all-ranges*. Jako dane wejściowe dla testu Friedmana posłużyły średnie wartości metryki  $m_1$  (wzór 4.4) uzyskane przez poszczególne algorytmy na 16 testowych zbiorach danych. Przeprowadzenie analizy statystycznej wszystkich 30 wariantów jednocześnie wymagałoby wykonaniu eksperymentów na wielu więcej zbiorach danych - w tym przypadku test Friedmana wykazuje różnice w algorytmach przy  $\alpha = 0.05$ , ale test Nemenyi pozwala na wyłonienie jednej grupy, zawierającej wszystkie warianty. Porównanie algorytmów przeprowadzono więc w podgrupach: metody nadzorowane tą samą miarą, metody pogrupowane po kierunku indukcji (F/B/FB), osobno modele akcyjne regułowe i oparte o algorytm rekomendacji. Dla każdej z podgrup w dalszej części rozdziału przedstawiono te, dla których wyniki testu Friedmana były istotne. Wyniki zostały przedstawione w postaci CD-diagramów na rysunkach 5.5, 5.6, 5.7 i 5.8 z wartością krytyczną CD wyznaczoną wg. wzoru 5.3,  $\alpha = 0.1$ .

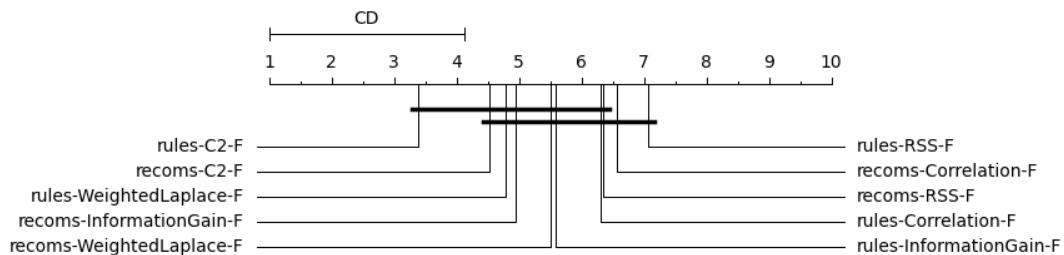
Test Nemenyi nie wykazał więcej niż jednej grupy tylko w przypadku fuzji metod z rodziny F i B, dla którego to grupy wyników diagram CD zaprezentowano na

## 5 Eksperymenty

rysunku 5.5. Być może dalsze obniżenie wartości  $\alpha$  dla tego testu pozwoliłoby wyłonić więcej grup, jednak dla spójności z pozostałymi wynikami, należy stwierdzić, że przy poziomie istotności  $\alpha = 0.1$  test Nemenyi pozwala nam jedynie stwierdzić, że wszystkie metody z rodziny FB nie są istotnie statystycznie różne, choć oczywiście - co również prezentuje narysowany diagram CD - można je uporządkować zgodnie z przypisanymi im rangami. Pozostałe diagramy CD informują nas, że test Nemenyi wykazał dwie grupy algorytmów. Głębsza analiza pozwala stwierdzić, że wśród algorytmów nadzorowanych miarą WeightedLaplace wyróżnia się metoda *rules-WeightedLaplace-FB*, zaś spośród wszystkich modeli akcyjnych zbudowanych w oparciu o zbiory reguł najlepszy okazuje się być wariant *rules-C2-FB*. Wśród metod opierających się o kierunek indukcji w przód wyróżnia się wariant *rules-C2-F*.



**Rysunek 5.5:** Diagram CD wskazujący na różnice w uzyskanej skuteczności transformacji przykładów między grupami modeli akcyjnych utworzonych z algorytmów regułowego i rekomendacji o kierunku indukcji *FB* (fuzja metod *F* i *B*).

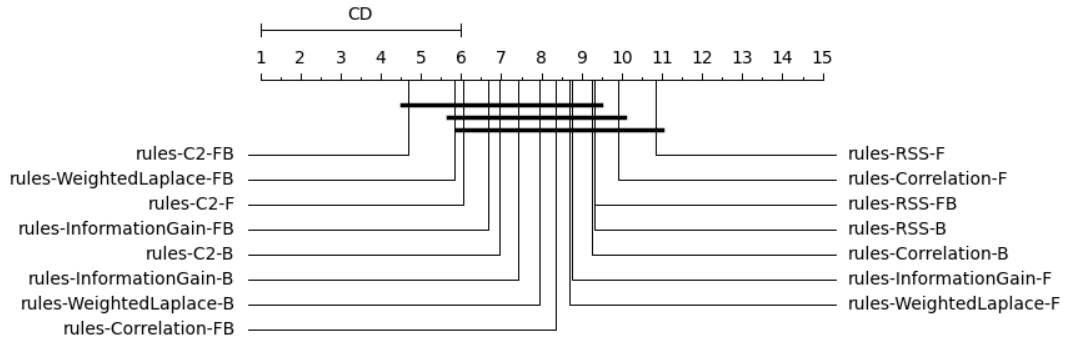


**Rysunek 5.6:** Diagram CD wskazujący na różnice w uzyskanej skuteczności transformacji przykładów między grupami modeli akcyjnych utworzonych z algorytmów regułowego i rekomendacji o kierunku indukcji *F*.

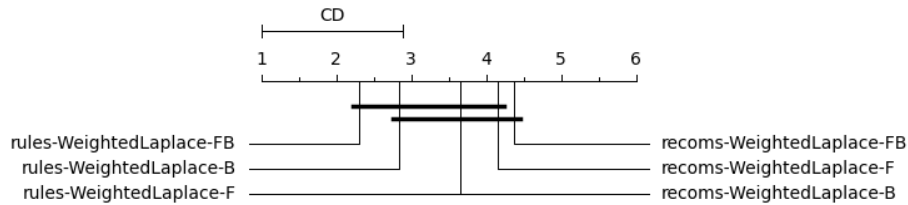
Analiza statystyczna skuteczności modeli akcyjnych przedstawionych w niniejszej pracy nie pokazuje jednoznacznie, że jedna z metod, lub że pewna grupa algorytmów



## 5 Eksperymenty



**Rysunek 5.7:** Diagram CD wskazujący na różnice w uzyskanej skuteczności transformacji przykładów między grupami modeli akcyjnych utworzonych na podstawie różnych wariantów algorytmu regułowego



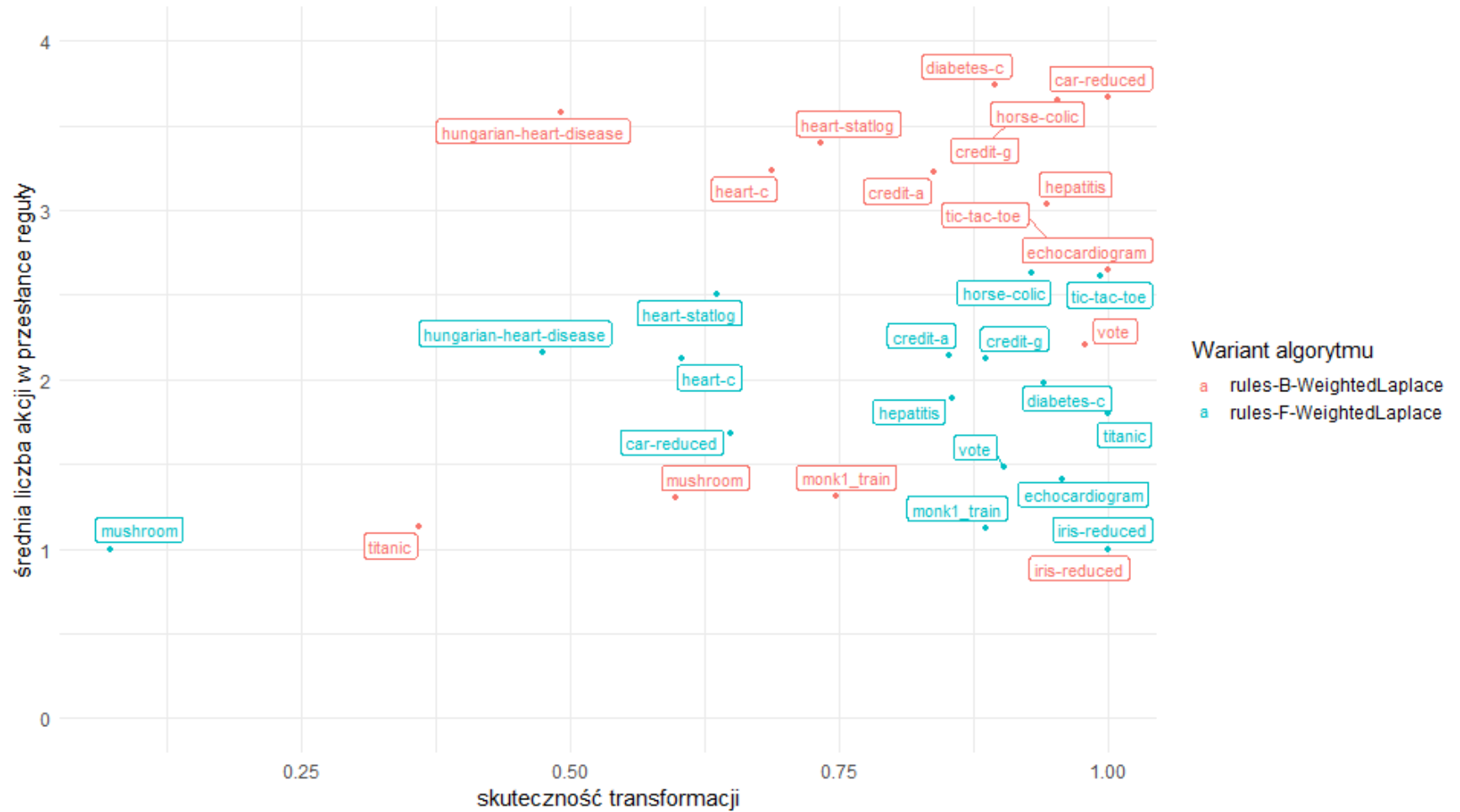
**Rysunek 5.8:** Diagram CD wskazujący na różnice w uzyskanej skuteczności transformacji przykładów między grupami modeli akcyjnych utworzonych z algorytmów regułowego i rekomendacji nadzorowanych miarą jakości WeightedLaplace

pozwała uzyskiwać wyniki znacznie lepsze od pozostałych. Można odnieść wręcz odwrotne wrażenie, że wykorzystany test statystyczny Friedmana z analizą post-hoc Nemenyi często nie pozwala wykryć różnic między poszczególnymi metodami. Podchodząc sceptycznie do tak zaprezentowanych wyników, można by stwierdzić, że przedstawione namnożenie metod jest niepotrzebne, gdyż zmiana parametrów budujących algorytmy nie prowadzi do istotnego ulepszenia uzyskiwanych wyników. Nie należy też zapominać, że skuteczność transformacji jest tylko jedną z cech warunkujących postrzeganie jakości modelu akcyjnego. Nie mniej istotna jest czytelność, reprezentacja wiedzy, jaką niesie ze sobą model akcyjny. Sugerowane zmiany wartości atrybutów przykładów są podawane w postaci reguł akcji, a parametry algorytmów takie jak kierunek indukcji czy miara nadzorująca mają istotny wpływ na ich postać, liczbę akcji i prostych warunków elementarnych w przesłance reguły, co zaś ma przeniesienie na czytelność i użyteczność odpowiedzi podawanej przez model akcji.

Sytuacja, w której różne warianty metody pozwalają uzyskać podobną skuteczność jest niekomfortowa dla chcącego wdrożyć metody uczenia maszynowego do rozwiązania zadanego problemu. W takim przypadku należy podjąć decyzję w oparciu o analizę wielokryterialną. Sama metryka  $m_1$  opisuje skuteczność modelu akcji, ale nie mówi nic o pozostałych, istotnych cechach modelu: o tym jak złożone są reguły, jak wiele akcji zawartych jest w przesłankach reguł, jak często spotyka się akcje puste. Wszystkie te cechy mogą mieć istotne znaczenia dla końcowego użytkownika budowanego rozwiązania.

Celem pokazania w jaki sposób można pogłębić analizę i dokonać wyboru najlepszego dla danego zastosowania wariantu, przeanalizujemy dokładniej diagram 5.8. W grupie algorytmów najbardziej skutecznych znalazły się trzy metody: *rules-WeightedLaplace-F*, *rules-WeightedLaplace-B* i *rules-WeightedLaplace-FB*, przy czym tylko ostatni z wymienionych wariantów nie został jednocześnie zakwalifikowany do innych grup. Przyjmijmy, że ze względu na większy rozmiar modelu akcji w wariancie *FB*, odrzucamy ten wariant z dalszej analizy.

Wykres przedstawiający zależność skuteczności transformacji (tj. wartości metryki  $m_1$ ) i średniej liczby akcji w przesłankach reguł przedstawiono na rysunku 5.9. Wykres posiada 32 punkty danych (16 zbiorów x 2 warianty). Z jego analizy możemy wyczytać, że wariant *rules-B-WeightedLaplace* dla większości analizowanych zbiorów generuje reguły zawierające więcej akcji elementarnych niż dzieje się to podczas indukcji reguł w przód. Płyne z tego informacja, że choć obie metody średnio uzyskują podobną skuteczność, indukcja w przód dostarcza mniej informacji o potrzebnych zmianach wartości atrybutów. Wyższa liczba akcji elementarnych może okazać się cechą pożądaną. W przeciwieństwie do algorytmów przedstawionych np. w pracach [44, 96], metody przedstawione w tej pracy nie biorą pod uwagę kosztu wykonania akcji. W rzeczywistości może zdarzyć się, że na pewne atrybuty rzeczywistych obiektów można wpływać znacznie niższym kosztem, niż na inne. Zaproponowane algorytmy nie są w stanie uwzględnić tej wiedzy w procesie indukcji reguł. Alternatywą może być zastosowanie miary, która generuje więcej akcji, opartych o większą liczbę atrybutów, i spośród tak odkrytych reguł – przy pomocy eksperta domenowego – wybrać te akcje, które są interesujące, a jednocześnie obciążone niskim kosztem.



**Rysunek 5.9:** Skuteczność transformacji przykładów  $m_1$  w relacji do średniej liczby akcji w przesłankach reguł dla dwóch wybranych wariantów algorytmu.

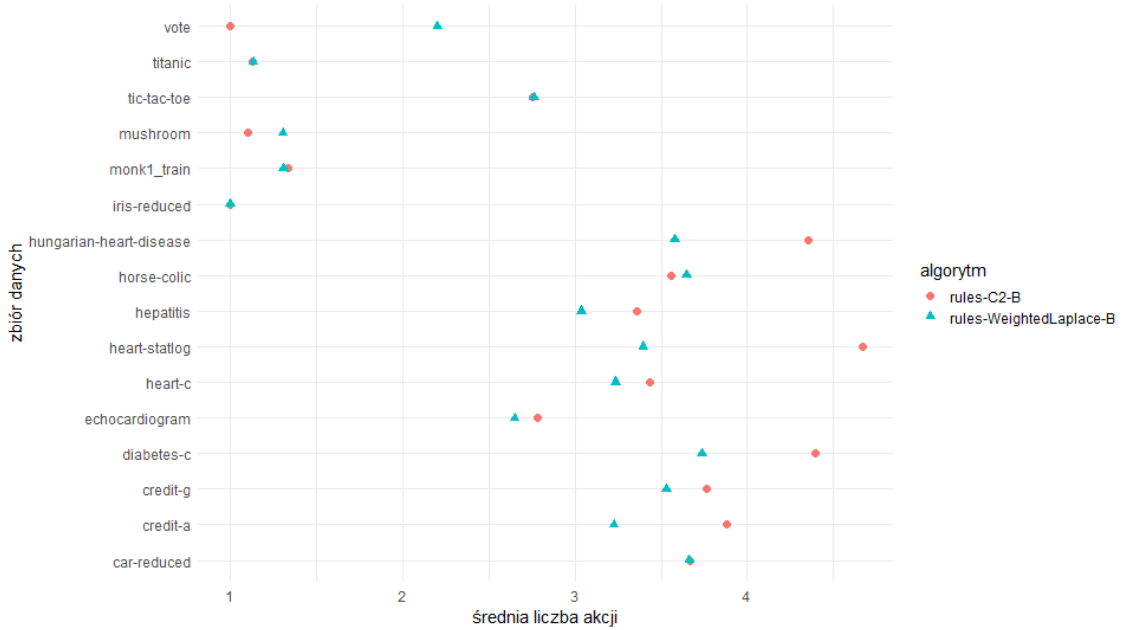
Przeprowadzona analiza statystyczna nie wykazuje jednej metody zdecydowanie przewyższającej inne w kontekście skuteczności transformacji (uzyskanych wartości metryki  $m_1$ ), jednak wyniki działania przedstawionych metod, choć cechujące się podobną skutecznością, różnią się odkrytą wiedzą na tematy możliwych akcji i złożonością jej prezentacji. Stosowanie różnych miar jakości nadzorujących proces wzrostu i przycinania reguł prowadzi do różnych pod kątem ilościowym i jakościowym zbiorów reguł a w konsekwencji do innych rekomendacji utworzonych na podstawie reguł. Na podstawie diagramu 5.7 widać, że najczęściej wyższe miejsca w rankingu uzyskiwały algorytmy nadzorowane miarami *C2*, *InformationGain* i *WeightedLaplace*. W celu wypuklenia różnic w reprezentacji wiedzy uzyskanych przez metody nadzorowane różnymi miarami jakości, w dalszej części zestawiono ze sobą wyniki uzyskane przez metody *rules-C2-B* i *rules-WeightedLaplace-B*. Nie są to metody, które uzyskały najwyższe miejsca w rankingu, a ich wybór jest podyktowany następującymi kryteriami: obie metody średnio zajęły wysokie miejsca w rankingu skuteczności, obie metody opierają się na tym samym kierunku indukcji i nie stanowią wariantu będącego fuzją wariantów F i B. W przypadku wariantów będących fuzją, analiza postaci uzyskanych zbiorów reguł jest wtórna do analizy składowych zbiorów reguł, uzyskanych niezależnie przez warianty F i B.

	rules-C2-B	rules-WeightedLaplace-B
średnia liczba reguł	15.87	20.45
średnia liczba akcji	3.00	2.82
średnia liczba warunków	3.57	3.27
średnia precyzja reguły źródłowej	0.92	0.91
średnia precyzja reguły docelowej	0.95	0.97

**Tabela 5.13:** Uśrednione na 16 zbiorach charakterystyki zbiorów reguł akcji uzyskanych przy użyciu wybranych wariantów algorytmu

W tabeli 5.13 zebrano parę uśrednionych charakterystyk zbiorów reguł akcji uzyskanych przez wybrane warianty algorytmu. Wykorzystanie miary *WeightedLaplace* do nadzorowania indukcji reguł akcji prowadzi do liczniejszych zbiorów reguł, które jednak zawierają w swoich przesłankach mniej akcji elementarnych, niż zbiory uzyskane przy użyciu miary *C2*. W kontekście precyzji reguł źródłowych i docelowych, oba warianty uzyskują podobny wynik, z niewielką przewagą przy wykorzystaniu miary *WeightedLaplace*. Na wykresach 5.11 i 5.10 przedstawiono, dla każdego z

analizowanych zbiorów, średnie liczby reguł akcji i akcji elementarnych zawartych w przesłankach tych reguł. Wykresy te potwierdzają wnioski płynące z analizy tabeli 5.13.



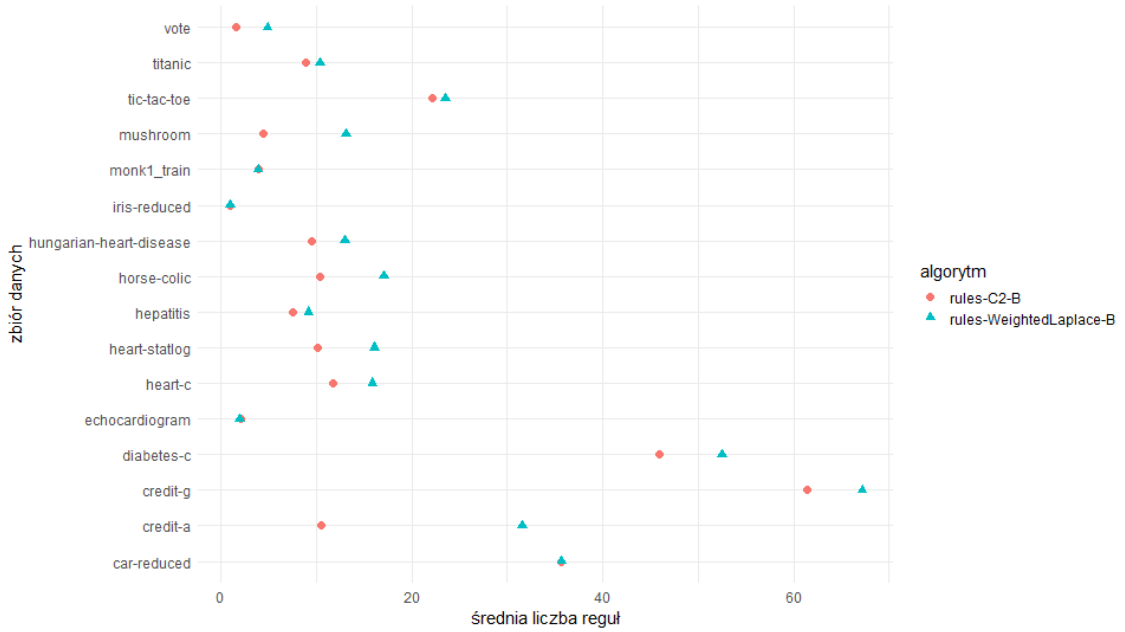
**Rysunek 5.10:** Średnia liczba akcji w przesłankach reguł akcji uzyskanych w 10 przebiegach wariantów *rules-C2-B* i *rules-WeightedLaplace-B*

Przyjrzyjmy się postaci uzyskanych reguł. Poniżej zamieszczono reguły uzyskane w wyniku działania wariantów *rules-C2-B* i *rules-WeightedLaplace-B* w pierwszym przebiegu sprawdzianu krzyżowego na zbiorze *mushroom*. Przykład 5.3.1 zawiera reguły odkryte przy użyciu miary C2. Wynik działania algorytmu *rules-WeightedLaplace-B* przedstawiono w przykładzie 5.3.2.

### Przykład 5.3.1

$$\begin{aligned}
 r1 &: \mathbf{IF} (odor, f \rightarrow n) \mathbf{THEN} (class, p \rightarrow e) \\
 r2 &: \mathbf{IF} (population, v \rightarrow n) \mathbf{THEN} (class, p \rightarrow e) \\
 r3 &: \mathbf{IF} (odor, f \rightarrow a) \mathbf{THEN} (class, p \rightarrow e) \\
 r4 &: \mathbf{IF} (odor, f \rightarrow l) \mathbf{THEN} (class, p \rightarrow e)
 \end{aligned} \tag{5.7}$$

## 5 Eksperymenty



**Rysunek 5.11:** Średnia liczba akcji uzyskana w 10 przebiegach wariantów *rules-C2-B* i *rules-WeightedLaplace-B*

### Przykład 5.3.2

- $r1 : \mathbf{IF} (stalk\text{-}color\text{-}below\text{-}ring, b \rightarrow g) \mathbf{THEN} (class, p \rightarrow e)$   
 $r2 : \mathbf{IF} (stalk\text{-}color\text{-}above\text{-}ring, b \rightarrow g) \mathbf{THEN} (class, p \rightarrow e)$   
 $r3 : \mathbf{IF} (population, v \rightarrow n) \mathbf{THEN} (class, p \rightarrow e)$   
 $r4 : \mathbf{IF} (odor, f \rightarrow a) \mathbf{THEN} (class, p \rightarrow e)$   
 $r5 : \mathbf{IF} (odor, f \rightarrow l) \mathbf{THEN} (class, p \rightarrow e)$   
 $r6 : \mathbf{IF} (population, v \rightarrow a) \mathbf{THEN} (class, p \rightarrow e)$   
 $r7 : \mathbf{IF} (stalk\text{-}color\text{-}above\text{-}ring, b \rightarrow o) \mathbf{THEN} (class, p \rightarrow e)$   
 $r8 : \mathbf{IF} (habitat, p \rightarrow w) \mathbf{THEN} (class, p \rightarrow e)$   
 $r9 : \mathbf{IF} (ring\text{-}type, l \rightarrow f) \mathbf{THEN} (class, p \rightarrow e)$   
 $r10 : \mathbf{IF} (stalk\text{-}shape, ANY \rightarrow t) \wedge (odor, f \rightarrow n) \mathbf{THEN} (class, p \rightarrow e)$   
 $r11 : \mathbf{IF} (spore\text{-}print\text{-}color, ANY \rightarrow n) \wedge (odor, f \rightarrow n) \mathbf{THEN} (class, p \rightarrow e)$   
 $r12 : \mathbf{IF} (spore\text{-}print\text{-}color, ANY \rightarrow k) \wedge (odor, f \rightarrow n) \mathbf{THEN} (class, p \rightarrow e)$   
 $r13 : \mathbf{IF} (odor, f \rightarrow n) \wedge (cap\text{-}color, ANY \rightarrow g) \mathbf{THEN} (class, p \rightarrow e)$
- (5.8)

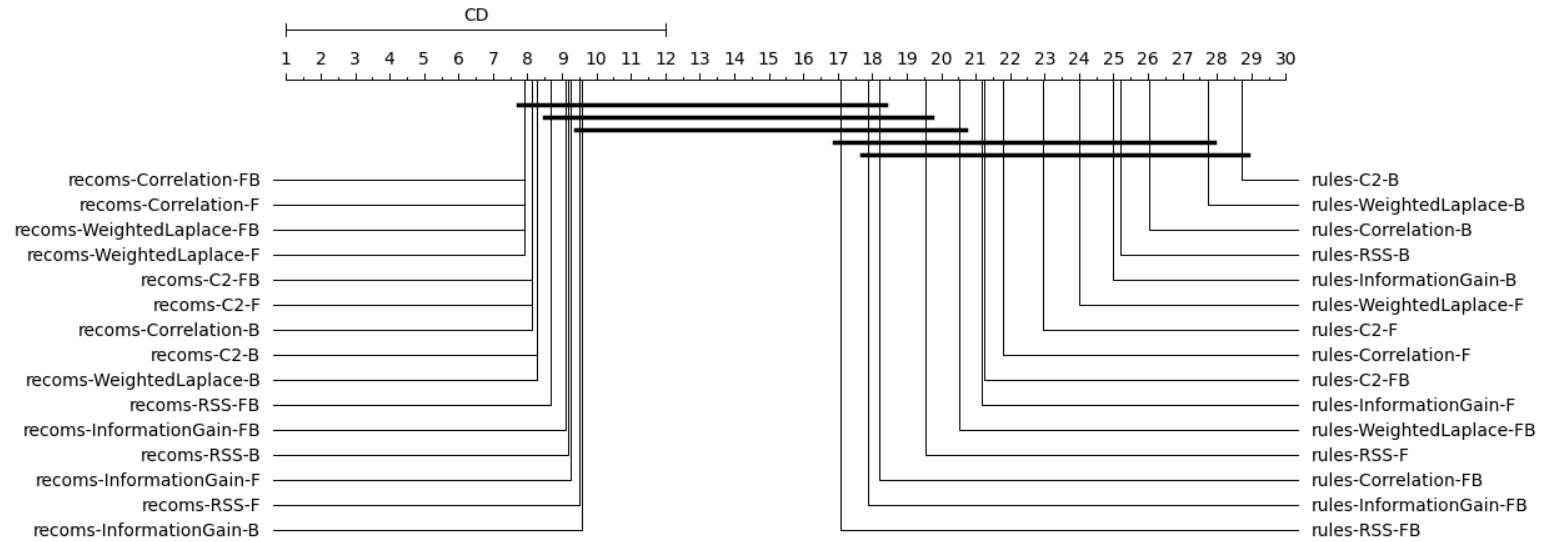
W przedstawionych wynikach pominięto informację o pokryciu reguł, jednak wymaga ona pewnego komentarza. Wariant wykorzystujący miarę C2 wypełnił wymagania minimalnego pokrycia zbioru przy użyciu zaledwie 4 reguł, z czego 2 reguły są regułami dokładnymi, zarówno po stronie źródłowej jak i docelowej. Pierwsza reguła w uzyskanym zbiorze pokrywa 2829 przykładów klasy docelowej (przy czym niepoprawnie klasyfikuje tylko 91 przykładów klasy docelowej), co przekłada się na pokrycie ok. 81% wszystkich przykładów klasy docelowej. 3 z 4 uzyskanych reguł zawierają w przesłance akcję opartą o atrybut *odor*. Wariant WeightedLaplace potrzebuje 13 reguł, wśród których znajdziemy 10 reguł dokładnych po obu stronach. Reguły charakteryzują się niższym pokryciem niż w przypadku C2, ale w przesłankach reguł pojawia się łącznie 9 różnych atrybutów, na których budowane są akcje.

Podsumowując, miara C2 pozwoliła uzyskać zwięzły zbiór reguł, ale różnorodność odkrytych akcji jest niska. Jak pokazała analiza skuteczności transformacji, nie stoi to na przeszkodzie w formułowaniu dobrych rekomendacji. Miara WeightedLaplace prowadzi do bardziej skomplikowanej reprezentacji regułowej, nie tylko pod kątem liczby reguł, ale także liczby warunków w ich przesłankach oraz różnorodności atrybutów wykorzystanych w akcjach elementarnych. Ponownie należy stwierdzić, że w obliczu podobnej skuteczności analizowanych wariantów, przy wyborze najlepszej metody trzeba kierować się wieloma kryteriami. W tym przypadku, większa różnorodność atrybutów pojawiających się w przesłankach reguł pozwala na bardziej elastyczne rekomendowanie pewnych działań, ponieważ użytkownik metody ma większy wybór dostępnych akcji. W szczególności indukcja rekomendacji na podstawie zbioru reguł o bardziej rozbudowanych przesłankach będzie prowadzić do rekomendacji mogących sugerować zmiany na większej liczbie atrybutów, co wynika z konstrukcji algorytmu (jednak w tej pracy nie podjęto się stosownej analizy wyników po tym kątem). Ta cecha może być istotna, gdy koszt wdrażania akcji jest wysoki, i pożądane jest odkrycie akcji opartych o wiele atrybutów, by wśród nich wybrać te prowadzące do akcji opartych o atrybuty o najniższym koszcie modyfikacji. Oczywiście możliwa jest też sytuacja odwrotna - gdy wariant prowadzący do mniej różnorodnych przesłanek nie stanowi problemu, bo koszt wprowadzania zmian w rzeczywistych obiektach jest taki sam, niezależnie od zmienianej cechy.

Innym istotnym parametrem jest zdolność modelu akcji do generowania rekomendacji dla nieznanymi jeszcze przykładów, czyli pokrycie uzyskane przez dany model. Naturalnie preferowany będzie taki model, który jest w stanie wygenerować rekomendację dla większej liczby przykładów. W celu zweryfikowania względnej jakości

poszczególnych modeli akcyjnych właśnie pod kątem uzyskanego pokrycia, przeprowadzono dodatkowy test statystyczny. Badaną zmienną było pokrycie uzyskane przez wszystkie badane metody na zbiorach testowych. Pokrycie zostało wyznaczone jako iloraz liczby wygenerowanych rekomendacji i liczby przykładów w zbiorze testowym. W tym przypadku nie wzięto pod uwagę, czy dana rekomendacja jest oceniona jako skuteczna czy nie, co byłoby rozpatrywane w przypadku analizy metryk 4.5 i 4.6. Działanie to ma na celu pokazanie, w jaki sposób można samodzielnie definiować istotne dla użytkownika metryki, i jak można ocenić metody indukcji rekomendacji względem tej metryki przed wyborem jednej z nich do zastosowania praktycznego. Test Friedmana na poziomie istotności  $\alpha = 0.05$  pozwala odrzucić hipotezę zerową, zaś test Nemenyi wyróżnia 5 istotnie różnych grup algorytmów, co prezentuje CD-diagram na rysunku 5.12. Łącząc informacje uzyskane z przedstawionego CD-diagramu z tymi, które uzyskaliśmy we wcześniejszych testach, możemy sformułować bardzo istotny wniosek: modele akcyjne zbudowane przy użyciu algorytmu indukcji rekomendacji charakteryzują się zdecydowanie lepszą zdolnością do generowania rekomendacji dla nieznanymi przykładów. Jest to bardzo ważne spostrzeżenie, gdyż spoglądając na samą analizę skuteczności poszczególnych algorytmów, moglibyśmy stwierdzić, że algorytm rekomendacji wprowadza pewną dodatkową złożoność, która nie daje oczekiwanych zysków. Nie analizując jakości modeli wyrażonych innymi metrykami, jak na przykład pokryciem, odrzucilibyśmy algorytm rekomendacji, podczas gdy może on się okazać niezastąpiony w zastosowaniach, w których konieczne jest uzyskanie możliwie największej liczby rekomendacji.





**Rysunek 5.12:** Diagram CD wskazujący na różnice w zdolności do pokrywania przykładów testowych między grupami modeli akcyjnych utworzonych z algorytmów regułowego i rekomendacji.

## 5.4 Reguły akcji w zadaniu analizy danych regresyjnych

W rozdziale 3.3.1 opisano sposób uzyskiwania reguł akcji w zadaniu regresji. W tym rozdziale przedstawiono i omówiono wyniki uzyskane przy użyciu procesu oceny jakości regresyjnych modeli akcji opisanego w rozdziale 4.3.4.

Regresyjne reguły akcji wykorzystane w tym rozdziale były indukowane przy użyciu programu realizującego algorytm 13. W trakcie indukcji reguł nie było nałożone żadne ograniczenie na relację między wartościami konkluzji reguły źródłowej i docelowej.

### 5.4.1 Charakterystyka zbiorów danych

W tabeli 5.14 zawarto zbiory, w których zmienna decyzyjna jest ciągła, a więc są odpowiednie do testowania reguł akcji w zadaniu regresji. Oprócz nazwy zbiorów, tabela zawiera kilka podstawowych charakterystyk takich jak liczba przykładów czy liczba typów atrybutów. W porównaniu do podobnej tabeli przedstawionej w podrozdziale 5.3.1 dotyczącym klasyfikacji, brak w podsumowaniu zbiorów wskazania na klasę źródłową i docelową. Wynika to oczywiście z faktu, że dla zadania regresji nie da się dokonać takiego podziału, a analizować będziemy skuteczność zastosowania reguł akcji i rekomendacji w odniesieniu do zmian wartości ciągłego atrybutu decyzyjnego. Zbiory *auto-mpg* i *houses\_taiwan* zostały pobrane z repozytorium UCI [23], zbiory *boston\_housing* i *bodyfat* udostępniono w portalu Kaggle<sup>1</sup>, zbiór *methane-train-minimal* pochodzi z przykładów zawartych w repozytorium RuleKit [39]<sup>2</sup>, natomiast zbiór *gas* to popularny zbiór używany do nauki predykcji ciągów czasowych [7] dostępny w sieci<sup>3</sup>, przetworzony podobnie jak w [18, 84]. Podobnie jak w przypadku doboru zbiorów dla zadania klasyfikacji, zbiory zostały dobrane pod kątem intuicyjności danych w nich zawartych, aby łatwo można było zweryfikować poprawność znaczenia uzyskanych reguł akcji (np. *houses\_taiwan*, *boston\_housing*) lub pod kątem ich popularności w innych pracach (*gas*, *methane-train-minimal*).

W rozdziale 5.3 rolę klasyfikatora weryfikującego pełnił algorytm XGBoost. Dla celów oceny jakości modeli akcyjnych w zadaniu regresji zastosowano ten sam algorytm, jednak z użyciem funkcji celu odpowiedniej dla zadania regresji: *reg:squarederror* (więcej detali w dokumentacji pakietu *xgboost* dla języka R [14]). Pozostałe parametry

<sup>1</sup><https://www.kaggle.com/>

<sup>2</sup><https://github.com/adaa-polsl/RuleKit>

<sup>3</sup><https://www.stat.purdue.edu/~chong/stat520/bjr-data/>

trenowania algorytmu, podobnie jak w zadaniu klasyfikacji, pozostały ustawione na wartości domyślne, z wyjątkiem maksymalnej liczby przebiegów ustawionej na 10000 i parametru *early\_stopping\_rounds* ustalonego na wartość 10, co przekłada się na przerwanie procesu trenowania, jeśli wartość funkcji celu nie polepszy się w 10 następujących po sobie przebiegach lub po 10000 przebiegów.

Tabela 5.15 przedstawia uśrednione wartości błędów średniokwadratowego i bezwzględnego uzyskane w procesie trenowania i testowania algorytmu XGBoost na testowych zbiorach danych z tabeli 5.14. Tam, gdzie wartość błędu RMSE lub MAE w tabeli 5.15 jest równa 0, oznacza to błąd bardzo niski i jest wynikiem zaokrąglenia do dwóch liczb.

Nazwa zbioru	#przykładów	#atrybutów	Atrybuty nominalne	Atrybuty numeryczne	Zakres zm. decyzyjnej
auto-mpg	398	7	3	4	45.0
bodyfat	252	15	0	15	47.5
boston_housing	506	13	1	12	45.0
gas	290	11	0	11	14.9
houses_taiwan	414	7	0	7	109.9
methane-train-minimal	200	8	0	8	1.2

**Tabela 5.14:** Zbiory danych wykorzystane w eksperymentach w zadaniu regresji i ich podstawowe charakterystyki.

Nazwa zbioru	RMSE	RMSE	MAE	MAE
	zb. treningowy	zb. testowy	zb. treningowy	zb. testowy
auto-mpg	0.00	3.15	0.00	2.23
bodyfat	0.00	1.83	0.00	0.73
boston_housing	0.00	3.15	0.00	2.23
gas	0.00	0.52	0.00	0.39
houses_taiwan	0.70	8.29	0.15	5.32
methane-train-minimal	0.01	0.09	0.00	0.05

**Tabela 5.15:** Średni błąd kwadratowy i średni błąd bezwzględny predykcji atrybutu decyzyjnego przez algorytm XGBoost. Wartości uśrednione z 10 przebiegów.

### 5.4.2 Indukcja reguł akcji i rekomendacji z danych regresyjnych

Eksperyment polegający na dziesięciokrotnym przeprowadzeniu procedury testowej opisanej w rozdziale 4.3 przeprowadzono na zbiorach danych opisanych w tabeli 5.14. Podstawowe, uśrednione charakterystyki zbiorów reguł akcji regresji uzyskanych z zastosowaniem różnych miar jakości nadzorujących proces indukcji reguł zebrano w tabeli 5.16.

Do najniższej liczby reguł w wynikowym zbiorze prowadzi zastosowanie miary InformationGain, do najwyższej zaś - RSS. Pozostałe miary prowadzą do, średnio, podobnych rozmiarów zbiorów wynikowych. Średnio najwięcej akcji, przy jednocześnie najmniejszej liczbie warunków elementarnych daje zastosowanie miary WeightedLaplace. Reguły o najdłuższych przesłankach znajdziemy w wynikach działania algorytmu nadzorowanego miarą Information Gain. Reguły o najwyższej precyzji po stronie źródłowej i docelowej daje miara C2. Analizując wyniki pod kątem pokrycia reguł, należy pamiętać o specyficznej cesze reguł regresji, dla których klasa pozytywna jest niejako płynna i zdeterminowana przez postać przesłanki reguły. Tym niemniej, poszukując zbiorów reguł o wysokich wartościach pokrycia w klasie źródłowej i docelowej, należy zwrócić się ku zastosowaniu miar InformationGain i RSS.

	C2	Correlation	Information Gain	RSS	Weighted Laplace
liczba reguł	25.02	23.37	15.88	28.68	25.08
liczba warunków elementarnych	4.23	4.12	4.60	3.94	2.46
liczba akcji elementarnych	1.31	1.20	1.25	1.22	1.41
precyzja części źródłowej	0.88	0.81	0.81	0.72	0.72
precyzja części docelowej	0.83	0.77	0.77	0.71	0.72
pokrycie części źródłowej	0.64	0.79	0.83	0.85	0.52
pokrycie części docelowej	0.47	0.75	0.80	0.80	0.41

**Tabela 5.16:** Wybrane charakterystyki zbiorów reguł akcji regresji, uśrednione na 6 zbiorach i 10 przebiegach.

W tabeli 5.17 przedstawiono charakterystyki zbiorów reguł uzyskane na każdym z testowanych zbiorów osobno przy użyciu miary WeightedLaplace, zaś w tabeli 5.18 przy użyciu miary InformationGain. Przykładowa reguła, odkryta na zbiorze *houses-taiwan* przy użyciu miary InformationGain prezentuje się następująco:

**IF** ( $\text{distance\_to\_station}, < 826.83, \infty) \rightarrow (-\infty, 589.41) \wedge$   
 $(\text{latitude}, < 24.95, 24.99) \rightarrow < 24.96, \infty) \wedge (\text{age}, (-\infty, 32.35) \rightarrow (-\infty, 37.65))$   
**THEN** ( $\text{class}, 25.90 \pm 5.47 \rightarrow 46 \pm 10.37$ )

Atrybut *distance\_to\_station* oznacza odległość nieruchomości od stacji metra w metrach, *age* to wiek nieruchomości w latach, zaś *latitude* to szerokość geograficzna położenia nieruchomości. Atrybut decyzyjny o nazwie *class* oznacza cenę (w lokalnej walucie) jednostki powierzchni nieruchomości. Prezentowana reguła wskazuje wpływ odległości od stacji kolejki miejskiej na cenę nieruchomości (im bliżej, tym wyższa cena), oraz pewien związek z ogólnym położeniem geograficznym. Z jakiegoś powodu, nieruchomości znajdujące się w szerokości geograficznej między równoleżnikami 24.95 a 24.96 charakteryzują się niższą ceną, lub też nieruchomości znajdujące się dalej niż równoleżnik 24.96 charakteryzują się wysoką ceną - niestety, bez znajomości warunków geograficznych i społeczno-ekonomicznych Tajwanu ciężko jest ocenić zasadność odkrytej zależności, choć oczywiście znalazła ona potwierdzenie w danych, na podstawie których wygenerowano powyższą regułę akcji. Przedstawiona reguła charakteryzuje się następującymi wartościami  $p_s = 64, n_s = 8, p_t = 133, n_t = 33$ .

zbiór	# reguł	# akcji	# warunków elementarnych	precyzja źródło	precyzja cel	pokrycie źródło	pokrycie cel
auto-mpg	15(4.24)	1.45(0.18)	2.83(0.18)	0.72(0.03)	0.71(0.03)	0.43(0.06)	0.14(0.05)
bodyfat	39(0)	1.04(0.08)	1.6(0.15)	0.67(0.02)	0.66(0.03)	0.89(0.02)	0.91(0.02)
boston_housing	15(4.24)	1.45(0.18)	2.83(0.18)	0.72(0.03)	0.71(0.03)	0.43(0.06)	0.14(0.05)
gas	44.9(0.32)	1.86(0.11)	2.89(0.09)	0.71(0.01)	0.71(0.02)	0.42(0.02)	0.41(0.03)
houses_taiwan	29.9(4.58)	1.51(0.2)	2.75(0.21)	0.74(0.03)	0.71(0.02)	0.3(0.03)	0.21(0.03)
methane-train-minimal	6.7(5.77)	1.14(0.33)	1.86(0.54)	0.76(0.08)	0.82(0.12)	0.61(0.13)	0.64(0.13)

**Tabela 5.17:** Charakterystyki zbiorów regresyjnych reguł akcji uzyskanych przy użyciu miary WeightedLaplace na poszczególnych zbiorach. Dane uśrednione z 10 przebiegów. W nawiasie podano odchylenie standardowe.

zbiór	# reguł	# akcji	# warunków elementarnych	precyzja źródło	precyzja cel	pokrycie źródło	pokrycie cel
auto-mpg	21.9(5.04)	1.44(0.43)	5.21(0.37)	0.79(0.02)	0.76(0.03)	0.72(0.03)	0.65(0.04)
bodyfat	18.2(2.04)	0.64(0.21)	5.09(0.36)	0.82(0.02)	0.76(0.02)	0.93(0.03)	0.98(0.02)
boston_housing	21.9(5.04)	1.44(0.43)	5.21(0.37)	0.79(0.02)	0.76(0.03)	0.72(0.03)	0.65(0.04)
gas	17.7(2.21)	1.28(0.26)	5.87(0.3)	0.78(0.01)	0.72(0.02)	0.86(0.04)	0.9(0.06)
houses_taiwan	7.8(2.04)	1.67(0.52)	3.6(0.43)	0.83(0.03)	0.8(0.04)	0.78(0.04)	0.74(0.13)
methane-train-minimal	7.8(1.62)	1.06(0.25)	2.6(0.51)	0.85(0.02)	0.81(0.03)	0.95(0.02)	0.88(0.06)

**Tabela 5.18:** Charakterystyki zbiorów regresyjnych reguł akcji uzyskanych przy użyciu miary InformationGain na poszczególnych zbiorach. Dane uśrednione z 10 przebiegów. W nawiasie podano odchylenie standardowe.

**Skuteczność transformacji**

Analizę skuteczności transformacji przykładów w zadaniu regresji rozpoczniemy od przyjrzenia się informacjom na temat pokrycia zbioru testowego zgromadzonym w tabeli 5.19. Termin „pokrycie” w tym przypadku oznacza procent przykładów ze zbioru testowego jaki został skutecznie przetransformowany (tj. znalazła się reguła akcji, której część źródłowa pokrywa badany przykład lub wygenerowana została dla niego rekomendacja).

zbiór	algorytm	C2	Correlation	Information Gain	RSS	Weighted Laplace
auto-mpg	rekomendacje	100.00	100.00	100.00	100.00	20.00
auto-mpg	reguły	59.41	75.15	91.88	80.30	18.71
bodyfat	rekomendacje	100.00	100.00	100.00	100.00	30.20
bodyfat	reguły	92.60	93.80	94.20	92.00	90.20
boston_housing	rekomendacje	100.00	100.00	100.00	100.00	20.00
boston_housing	reguły	59.41	75.15	91.88	80.30	18.71
gas	rekomendacje	100.00	100.00	100.00	100.00	0.00
gas	reguły	93.62	94.83	93.62	94.83	91.90
houses_taiwan	rekomendacje	100.00	100.00	100.00	100.00	0.00
houses_taiwan	reguły	40.72	40.12	39.28	44.34	40.84
methane-train-minimal	rekomendacje	100.00	100.00	100.00	100.00	80.00
methane-train-minimal	reguły	79.00	85.75	86.25	89.00	27.75

**Tabela 5.19:** Średnie pokrycie zbioru testowego w trakcie oceny jakości modelu akcyjnego, dane w procentach, uśrednione z 10 przebiegów.

Podobnie jak w przypadku zadania klasyfikacji, algorytm rekomendacji radzi sobie dużo lepiej w podawaniu akcji dla wcześniej nieznanymi przykładów. Uwagę zwracają odstające wyniki dla obu algorytmów dla miary WeightedLaplace. Analizując cechy ilościowe zbiorów reguł uzyskanych przy użyciu różnych miar, WeightedLaplace zdawało się dawać zbiory o relatywnie krótkich przesłankach z dużą ilością akcji. Jak pokazuje analiza pokrycia, nie przełożyło się to jednak na przydatność takich reguł w zadaniu transformowania przykładów. Informację o skuteczności pokrywania przykładów w zbiorze testowym należy zawsze brać jako dodatkowy czynnik oceny



jakości modelu. Zaproponowane w rozdziale 4.3.4 metryki  $mr_1$  i  $mr_2$  nie biorą tego faktu pod uwagę, informacja o pokryciu jest w pewien sposób uwzględniona tylko w wartości metryki  $mr_3$ , której uśrednione wartości dla każdego zbioru testowego przedstawiono w tabeli 5.20.

zbiór	algorytm	C2	Correlation	Information Gain	RSS	Weighted Laplace
auto-mpg	rekomendacje	89.90	88.61	73.07	88.02	59.90
auto-mpg	reguły	66.33	48.22	57.00	48.71	58.20
bodyfat	rekomendacje	66.80	61.00	28.60	64.60	44.37
bodyfat	reguły	51.19	41.36	22.72	64.57	72.28
boston_housing	rekomendacje	89.90	88.61	73.07	88.02	59.90
boston_housing	reguły	66.33	48.22	57.00	48.71	58.20
gas	rekomendacje	91.72	77.59	76.21	86.55	—
gas	reguły	68.32	65.09	59.30	58.55	81.43
houses_taiwan	rekomendacje	67.23	56.87	56.75	57.35	—
houses_taiwan	reguły	39.64	30.63	26.99	34.51	54.87
methane-train-minimal	rekomendacje	64.25	45.00	61.00	18.50	65.00
methane-train-minimal	reguły	50.00	44.31	42.32	53.37	40.54

**Tabela 5.20:** Uśredniona z 10 przebiegów wartość metryki  $mr_3$ . Wartości brakujące, gdy dany wariant nie pokrył żadnego przykładu.

Łączna analiza tabel 5.19 i 5.20 pokazuje, że na ogół algorytm rekomendacji jest lepszy w zdolności przetransformowania przykładów testowych (tj. charakteryzuje się wyższym pokryciem), ale także wytworzone przez niego akcje są częściej skuteczne, niż ma to miejsce w przypadku algorytmu regułowego.

W dalszym ciągu należy pochylić się nad skalą „błędu” wprowadzanego przez model akcji w procesie transformowania przykładów. W rozdziale 4 dokonano niejako przededefiniowania znaczenia błędu w kontekście testowania modeli akcyjnych. Jako błąd mierzyć będziemy rozbieżność między predykcją wartości zmiennej decyzyjnej przez algorytm weryfikujący (XGBoost) a wartością wskazywaną przez regułę akcji lub rekomendację dla danego przykładu. Dopuszczalne wartości tak zdefiniowanego błędu muszą oczywiście być w trakcie analizy odniesione do skali zmiennej decyzyjnej

danego zbioru danych. W tabeli 5.14 znajduje się kolumna „Zakres zm. decyzyjnej” wskazująca na rozpiętość wartości atrybutu decyzyjnego w całym zbiorze, zaś tabela 5.15 wykazuje wartości błędów RMSE i MAE (uśrednione) na zbiorach treningowych i testowych wyznaczonych z analizowanych zbiorów. Opierając się o te dane, po pierwsze jesteśmy w stanie określić dokładność algorytmu weryfikującego, a po drugie jesteśmy w stanie odnieść wartość błędu do zakresu zmiennej decyzyjnej.

Tabele 5.21 i 5.22 zawierają wyznaczone wartości błędów RMSE i MAE. W obydwu tabelach, w znakomitej większości przypadków, niezależnie od zastosowanej miary, algorytm rekomendacyjny uzyskiwał niższe wartości obydwu rodzajów błędów. Szczególną uwagę należy zwrócić na wyniki uzyskane z zastosowaniem miary WeightedLaplace, które to wyznaczone są na podstawie na ogół małych zbiorów przykładów przetransformowanych, w szczególności, w dwóch przypadkach - zbiorów pustych, które powstały w wyniku nie pokrycia przez model akcji ani jednego przykładu testowego.

Ogólną przewagę algorytmu rekomendacyjnego można tłumaczyć tym, że decyzję o wartości zmiennej decyzyjnej wskazywanej przez rekomendację podejmuje się na podstawie bazy danych w postaci meta-tablicy, która to jest kompilacją informacji z wszystkich uzyskanych na danym zbiorze reguł akcji. W przypadku metody regułowej, decyzja jest podejmowana na podstawie jednej tylko reguły.

Ostatnim krokiem analizy uzyskanych wyników jest próba wskazania najlepszej metody przy użyciu analizy statystycznej uzyskanych wyników. Przeprowadzono test Friedmana (wariant Imana-Davenporta) dla wyników w trzech grupach: wszystkie algorytmy razem, tylko algorytmy regułowe, tylko algorytmy rekomendacji. Analizowaną zmienną była wartość błędu MAE. Test został przeprowadzony na poziomie istotności  $\alpha = 0.05$ .

Test nie wykazał istotnych różnic między algorytmami regułowymi. W przypadku algorytmów rekomendacyjnych, i wszystkich algorytmów na raz, test pozwolił odrzucić hipotezę zerową zakładającą równość algorytmów w danej grupie, w związku z czym przeprowadzono analizę post-hoc w procedurze Nemenyi i wyznaczono diagramy CD (poziom istotności  $\alpha = 0.1$ ), zamieszczone dalej, odpowiednio diagram 5.13 przedstawia ranking wszystkich metod, a diagram 5.14 zawiera ranking wariantów algorytmu rekomendacji.

Analizując przedstawione diagramy CD należy zwrócić uwagę na dwa istotne fakty. Po pierwsze, zazwyczaj rysując diagram CD, na pierwszym miejscu umieszcza się algorytm, który najczęściej uzyskał najwyższą wartość analizowanej zmiennej. W

zbiór	algorytm	C2	Correlation	Information Gain	RSS	Weighted Laplace
auto-mpg	rekomendacje	4.96	5.73	8.86	6.09	2.09
auto-mpg	reguły	11.59	14.51	10.87	14.12	15.96
bodyfat	rekomendacje	7.84	9.90	14.54	8.04	5.62
bodyfat	reguły	13.89	14.54	13.49	11.32	11.93
boston_housing	rekomendacje	4.96	5.73	8.86	6.09	2.09
boston_housing	reguły	11.59	14.51	10.87	14.12	15.96
gas	rekomendacje	1.37	2.22	2.17	1.69	0.00
gas	reguły	3.32	3.51	3.37	3.70	2.78
houses_taiwan	rekomendacje	14.43	17.20	16.60	16.47	0.00
houses_taiwan	reguły	17.22	18.39	17.23	15.81	16.41
methane-train-minimal	rekomendacje	0.32	0.43	0.32	0.63	0.26
methane-train-minimal	reguły	0.37	0.41	0.40	0.40	0.36

**Tabela 5.21:** Uśredniona z dziesięciu przebiegów wartość błędu RMSE (metryki  $mr_1$ ) modeli akcyjnych: regułowego i rekomendacyjnego.

przypadku diagramów 5.13 i 5.14 jest jednak inaczej - ponieważ w przypadku błędów MAE i RMSE interesujące jest uzyskanie jak najniższej wartości, najbardziej z lewej strony umieszczono algorytmy które uzyskały najwyższą rangę. Po przeciwnej stronie diagramu znajduje się algorytm o najwyższej randze, a więc - średnio - najczęściej uzyskujący najwyższą wartość odpowiedniego błędu. Po drugie, na pierwszym miejscu obu diagramów znalazł się wariant *recoms-WeightedLaplace*. Wiedząc o niskim pokryciu uzyskiwanym przez ten model, z którego wynika dwukrotne uzyskanie zerowego błędu MAE na skutek braku podjęcia jakiegokolwiek akcji, nie można bezkrytycznie przyjąć go jako najlepszy wariant. Procedura Nemenyi nie wyłoniła jednak tego wariantu jako samodzielnego lidera, ale jako członka liczniejszej grupy. W obydwu grupach na drugim miejscu rankingu znajduje się wariant *recoms-C2*, który charakteryzuje się bardzo wysokim pokryciem na wszystkich zbiorach testowych. Niestety, jednocześnie wariant ten w obu przypadkach jest członkiem więcej niż jednej grupy. Na tej podstawie można zalecić stosowanie miary C2 i algorytmu rekomendacji w problemach regresyjnych, jednak przed wdrożeniem modelu należałoby przeprowadzić

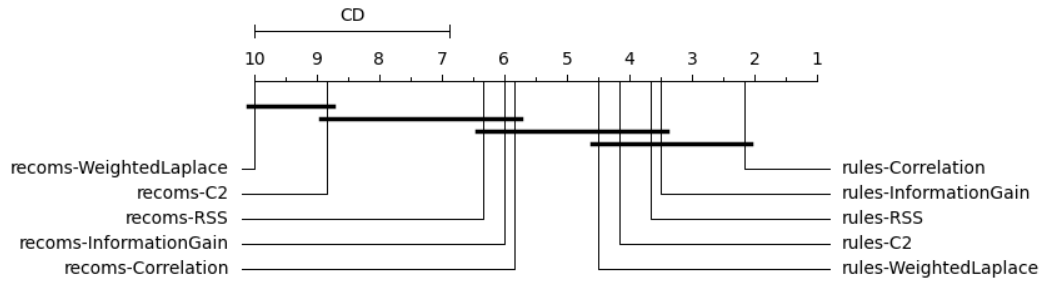
## 5 Eksperymenty

zbiór	algorytm	C2	Correlation	Information Gain	RSS	Weighted Laplace
auto-mpg	rekomendacje	3.72	4.12	7.02	4.22	1.74
auto-mpg	reguły	9.16	12.43	8.72	11.92	13.56
bodyfat	rekomendacje	6.07	8.19	13.16	6.47	5.00
bodyfat	reguły	10.62	11.69	12.36	8.21	8.06
boston_housing	rekomendacje	3.72	4.12	7.02	4.22	1.74
boston_housing	reguły	9.16	12.43	8.72	11.92	13.56
gas	rekomendacje	1.08	1.87	1.91	1.37	0.00
gas	reguły	2.58	2.86	2.75	3.05	2.04
houses_taiwan	rekomendacje	11.23	13.74	13.23	13.26	0.00
houses_taiwan	reguły	14.60	15.60	15.18	13.70	13.39
methane-train-minimal	rekomendacje	0.27	0.37	0.25	0.57	0.21
methane-train-minimal	reguły	0.32	0.35	0.35	0.34	0.31

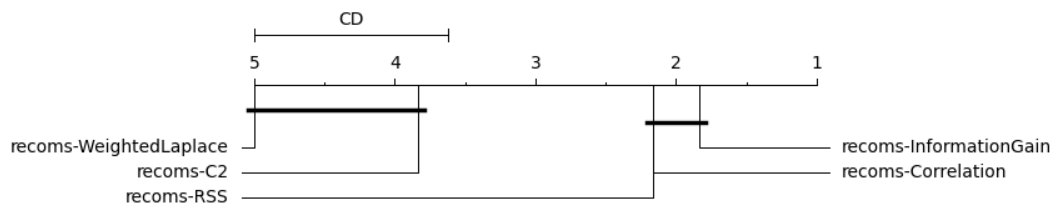
**Tabela 5.22:** Uśredniona z dziesięciu przebiegów wartość błędu MAE (metryki  $mr_2$ ) modeli akcyjnych: regułowego i rekomendacyjnego.

więcej testów na danych specyficznych dla danego zagadnienia celem potwierdzenia oczekiwanej skuteczności modelu akcyjnego.

## 5 Eksperymenty



**Rysunek 5.13:** Diagram CD wartości MAE dla wszystkich algorytmów w zadaniu regresji.



**Rysunek 5.14:** Diagram CD wartości MAE dla wszystkich wariantów algorytmu rekomendacyjnego w zadaniu regresji.

## 6 Podsumowanie

W niniejszej pracy przedstawiono ogólny zarys algorytmu indukcji reguł akcji z zastosowaniem paradygmatu sekwencyjnego pokrywania. Poprzez dobieranie kierunku indukcji i miary jakości nadzorującej indukcję reguł, użytkownik określający te parametry niejako generuje nowe algorytmy. Szereg tak opracowanych algorytmów został eksperymentalnie przetestowany na danych klasyfikacyjnych i opisujących problemy regresyjne. Testowanie polegało na ocenie skuteczności modeli akcyjnych utworzonych z uzyskanych zbiorów reguł lub rekomendacji zgodnie ze schematem postępowania również przedstawionym w tej pracy.

Wśród modeli akcyjnych pod kątem skuteczności wyróżniają się te, które uzyskane zostały z zastosowaniem miar *C2* i *WeightedLaplace*. Zastosowanie fuzji przedstawionych wariantów F i B ma wpływ na wzrost skuteczności w porównaniu do zastosowania wariantu F lub B osobno, lecz niesie ze sobą koszt obliczeniowy dwukrotnej indukcji reguł.

Indukcja reguł akcji z perspektywy klasy źródłowej (algorytmy z rodziny wariantu F) daje w wyniku reguły o cechach jakościowych i ilościowych innych niż w przypadku indukcji z punktu widzenia klasy docelowej, czyli metod z wariantu B, co może mieć znaczenie w przypadku wykorzystania podanych algorytmów w etapie odkrywania wiedzy. Analizując skuteczność transformacji tych dwóch rodzin algorytmów na zagregowanych wynikach z wielu zbiorów testowych nie wykazano znaczącej przewagi żadnego z wariantów. Być może przeprowadzono badania na zbyt małej liczbie zbiorów, lub wybrane zbiory testowe tworzą zestaw, na którym nie będziemy w stanie zaobserwować przewagi żadnego z wariantów. Tym niemniej, w zastosowaniach praktycznych dla ustalonego zbioru danych, wybór wariantu F lub B będzie prowadzić do zbiorów reguł o różnych charakterystykach, których skuteczność w zadaniu transformacji przykładów musi zostać zbadana przed wdrożeniem modelu. Przedstawiony w tej rozprawie sposób badania skuteczności modeli akcyjnych może łatwo zostać dostosowany do specyficznych wymagań użytkownika, poprzez podanie klasyfikatora weryfikującego właściwego dla analizowanych danych. Wybór najlepszej

kombinacji parametrów takich jak kierunek indukcji i miara nadzorująca nie powinien nastąpić problemów analitykowi chcącemu zastosować reguły akcji - wystarczy wykorzystać podany proces testowania modeli akcyjnych do znalezienia wariantu najskuteczniejszego.

Opisano również algorytm rekomendacji, będący swego rodzaju pogłębieniem procesu odkrywania wiedzy. Wejściem algorytmu nie są jednak surowe dane, a wcześniej odkryte reguły akcji. Podany algorytm rekomendacyjny buduje wewnętrzną bazę danych, będącą kompilacją wiedzy odkrytej już przez reguły akcji. Algorytm ten, w porównaniu do zastosowania wprost zbiorów reguł akcji, uwzględnia i adresuje problem konfliktów wśród reguł i faktu, że nierzadko przykład testowy może być pokryty przez więcej niż jedną regułę. W efekcie algorytm pozwala uzyskać specjalizowane pod zadany przykład reguły akcji. Jak wykazały przeprowadzone eksperymenty, dzięki swej konstrukcji algorytm rekomendacyjny lepiej radzi sobie z pokrywaniem przykładów ze zbioru testowego, a więc nie biorących udziału w procesie odkrywania reguł akcji.

Analiza przeprowadzona w rozdziale 5 tej pracy wskazuje, że najwyższą skuteczność transformacji uzyskuje się dla algorytmów nadzorowanych miarami C2 i WeightedLaplace, w przypadku danych klasyfikacyjnych. Prawidłowość ta jest widoczna dla wszystkich rodzin: indukcji „w przód”, „wstecz” oraz fuzji tych wariantów. W przypadku danych regresyjnych, po uwzględnieniu wszystkich opisanych szczegółów, najlepiej spisują się algorytmy nadzorowane miarą C2. Pod kątem zdolności do pokrywania przykładów testowych metoda rekomendacji przewyższa metodę regułową zarówno w przypadku danych klasyfikacyjnych jak i regresyjnych.

Wczesne wersje opisywanych w tej pracy algorytmów przedstawione zostały w artykułach [57] i [58]. Algorytm rekomendacji i sposób weryfikacji modeli akcyjnych opisano w [83]. Wprowadzone w tej pracy algorytmy zostały zaimplementowane jako część pakietu RuleKit [39], dostępnego publicznie, jako otwarte oprogramowanie, w serwisie Github (<https://github.com/adaa-polsl/SCARI>).

### 6.1 Kierunki rozwoju

Bez wątpienia niniejsza praca nie wyczerpuje tematu indukcji reguł akcji z zastosowaniem paradygmatu sekwencyjnego pokrywania. Pewne metody i optymalizacje procesu indukcji reguł klasyfikacyjnych, po przeniesieniu na grunt indukcji reguł akcji, mogą przynieść usprawnienie zaproponowanych metod.

W algorytmach indukcji reguł decyzyjnych oprócz zastosowanego w metodach zaproponowanych w niniejszej pracy procesu przycinania reguł, stosuje się także krok filtracji [72, 81]. Filtracja polega na usuwaniu ze zbioru tych reguł, które pod względem przyjętego kryterium jakości są uznane za niepotrzebne. Jest więc to forma przycinania zbioru reguł, która nie zmienia postaci poszczególnych reguł. Interesujące byłoby zbadanie wpływu filtracji reguł akcji na skuteczność modelu w transformowaniu przykładów. Badanie takie można by poprzedzić analizą częstości użycia reguł akcji do transformowania przykładów i próbą zdefiniowania takiego kryterium, które pozwoliłoby odrzucić reguły niebiorące udziału w tym procesie. Potencjalnie pozwoliłoby to na uzyskanie zbiorów reguł akcji o mniejszym rozmiarze przy zachowaniu ich skuteczności, ale też - potencjalnie - mniejszej zdolności do generalizacji.

Reguły akcji mogą znaleźć zastosowanie także w analizie ciągów czasowych. Nie trudno wyobrazić sobie praktyczne zastosowania w obszarach konserwacji predykcyjnej czy analizie danych ekonomicznych. Algorytm pokryciowy z powodzeniem odkrywa reguły akcji w zadaniach regresji, klasyfikacji i analizy przeżycia, można więc się spodziewać podobnych rezultatów dla analizy ciągów czasowych.

W pracy przedstawiono wyniki eksperymentalne uzyskane na podstawie pięciu miar jakości, których wybór był kierowany wcześniejszymi badaniami wskazującymi na ich skuteczność w indukcji reguł klasyfikacyjnych i innych [104]. Pożytecznym byłoby określenie jakości modeli akcji uzyskanych przy użyciu szerszego zestawu miar jakości. Nie ma gwarancji, że miary dające dobre wyniki w „klasycznych” zadaniach przełożą się na dobre wyniki w odkrywaniu akcji. Tezę tę warto zweryfikować doświadczalnie. W przedstawionych w pracy algorytmach ta sama miara nadzoruje proces wzrostu (specjalizacji) i przycinania reguł. W [104] do indukcji reguł decyzyjnych zastosowano kombinacje miar: jedna miara nadzoruje wzrost reguły, a inna jej przycinanie. Interesującym byłoby zbadanie, czy podobne podejście do indukcji reguł akcji pozwala uzyskać wyższą skuteczność.

Sposób oceny jakości modeli akcji zaproponowany w niniejszej pracy nie był do tej pory stosowany w szerokim zakresie, w szczególności nie zostało wykonane porównanie jakości modeli akcyjnych wygenerowanych przez inne, dostępne metody indukcji reguł akcji jak ARED, ARoGS, MARFS1/2 czy DAKAR2 (por. rozdział 2.5). Publikacja swoistego benchmarku metod indukcji reguł akcji z zastosowaniem zaproponowanej metody oceny, przyjmując jej popularyzację, mogłaby doprowadzić do zwiększonej aktywności w obszarze dalszego rozwoju metod odkrywania akcji, ale



przede wszystkim pozwoliła określić czy algorytm pokryciowy pozwala na uzyskiwanie reguł lepszych niż inne podejścia.

Pokryciowy algorytm indukcji reguł decyzyjnych pozwala na wykorzystanie w procesie odkrywania reguł znanej już wiedzy, wprowadzanej przez eksperta domenowego. Uzyskiwane zbiory reguł charakteryzują się wyższą skutecznością, niż odkrywane w pełni automatycznie [82]. Podobne podejście mogłoby znaleźć zastosowanie w odkrywaniu reguł akcji.

# Bibliografia

- [1] Agrawal, R., Srikant, R. Fast algorithms for mining association rules. Bocca, J. B., Jarke, M., Zaniolo, C., redaktorzy, *Proceedings of 20th International Conference on Very Large Data Bases, VLDB*, strony 487–499. Morgan Kaufmann, 1994.
- [2] An, A., Cercone, N. Rule quality measures for rule induction systems: Description and evaluation. *Computational Intelligence*, 17:409–424, 08 2001.
- [3] Andersen, T., Martinez, T. NP-completeness of minimum rule sets. *Proceedings of the 10th International Symposium on Computer and Information Sciences*, strony 411–418, 12 1995.
- [4] Bagavathi, A., Mummoju, P., Tarnowska, K., Tzacheva, A., Raś, Z. SARGs method for distributed actionable pattern mining using spark. strony 4272–4281, 12 2017.
- [5] Batra, R., Rehman, M. Actionable knowledge discovery for increasing enterprise profit, using domain driven-data mining. *IEEE Access*, PP:1–1, 12 2019.
- [6] Błaszczczyński, J., Stefanowski, J., Zając, M. Ensembles of abstaining classifiers based on rule sets. Rauch, J., Raś, Z. W., Berka, P., Elomaa, T., redaktorzy, *Foundations of Intelligent Systems*, strony 382–391, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.
- [7] Box, G., Jenkins, G. M. *Time Series Analysis: Forecasting and Control*. Holden-Day, 1976.
- [8] Brank, J., Mili-Frayling, N., Grobelnik, M., Mladenić, D. Training text classifiers with SVM on very few positive examples. *Technical report. MSR-TR-2003-34*, 05 2003.
- [9] Breiman, L. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

- [10] Bruha, I. Quality of decision rules: Definitions and classification schemes for multiple rules. 07 2021.
- [11] Bruha, I., Tkadlec, J. Rule quality for multiple-rule classifier: Empirical expertise and theoretical methodology1. *Intell. Data Anal.*, 7:99–124, 05 2003.
- [12] Cendrowska, J. Prism: An algorithm for inducing modular rules". *International Journal of Man-Machine Studies*, 27:349–370, 1987.
- [13] Chen, T., Guestrin, C. XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16*, strona 785–794, New York, NY, USA, 2016. Association for Computing Machinery.
- [14] Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., Mitchell, R., Cano, I., Zhou, T., Li, M., Xie, J., Lin, M., Geng, Y., Li, Y. *xgboost: Extreme Gradient Boosting*, 2020.
- [15] Clark, P., Niblett, T. The CN2 induction algorithm. *Machine Learning*, 3(4):261–283, Mar 1989.
- [16] Cohen, W. Fast effective rule induction. *Twelfth International Conference on Machine Learning: 1995*, 95, 10 2000.
- [17] Cohen, W. W. Fast effective rule induction. *In Proceedings of the Twelfth International Conference on Machine Learning*, strony 115–123. Morgan Kaufmann, 1995.
- [18] Czogała, E., Łęski, J. M. Fuzzy and neuro-fuzzy intelligent systems. *Studies in Fuzziness and Soft Computing*, 2000.
- [19] Dardzińska, A. *Action Rules Mining*, wolumen 468 serii *Studies in Computational Intelligence*. Springer, 2013.
- [20] Dean, J., Ghemawat, S. Mapreduce: Simplified data processing on large clusters. wolumen 51, strony 137–150, 01 2004.
- [21] Dembczyński, K., Kotłowski, W., Słowiński, R. ENDER: a statistical framework for boosting decision rules. *Data Min. Knowl. Discov.*, 21:52–90, 01 2010.

- [22] Demšar, J. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 01 2006.
- [23] Dheeru, D., Karra Taniskidou, E. UCI machine learning repository, 2017.
- [24] Duch, W., Jankowski, N., Grabczewski, K., Adamczak, R. Optimization and interpretation of rule-based classifiers. *strony* 1–13, 05 2000.
- [25] Frank, E., Hall, M. A., Holmes, G., Kirkby, R., Pfahringer, B., Witten, I. H. *Weka: A machine learning workbench for data mining.*, *strony* 1305–1314. Springer, Berlin, 2005.
- [26] Freitas, A. On rule interestingness measures. *Knowledge-Based Systems*, 12(5):309–315, 1999.
- [27] Freund, Y., Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.
- [28] Friedman, J. H., Popescu, B. E. Predictive learning via rule ensembles. *The Annals of Applied Statistics*, 2(3):916–954, 2008.
- [29] Friedman, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32:675–701, 1937.
- [30] Friedman, M. A comparison of alternative tests of significance for the problem of  $m$  rankings. *Annals of Mathematical Statistics*, 11:86–92, 1940.
- [31] Fürnkranz, J. Separate-and-conquer rule learning. *Artif. Intell. Rev.*, 13(1):3–54, Luty 1999.
- [32] Fürnkranz, J., Gamberger, D., Lavrač, N. *Foundations of Rule Learning*. Cognitive Technologies. Springer Berlin Heidelberg, 2012.
- [33] Fürnkranz, J., Flach, P. ROC'n'Rule learning - towards a better understanding of covering algorithms. *Machine Learning*, 58:39–77, 01 2005.
- [34] Geffner, H., Wainer, J. Modeling action, knowledge and control. *In Proceedings ECAI-98*, *strony* 532–536. John Wiley & Sons, 1998.

- [35] Geng, L., Hamilton, H. Interestingness measures for data mining: A survey. *ACM Comput. Surv.*, 38, 09 2006.
- [36] Goodfellow, I., Bengio, Y., Courville, A. *Deep Learning*. MIT Press, 2016.
- [37] Gruca, A., Sikora, M. Identification of the compound subjective rule interestingness measure for rule-based functional description of genes. strony 125–134, 09 2012.
- [38] Grzymala-Busse, J., Pei, C., Wang, B. Classification methods in rule induction. 01 1996.
- [39] Gudyś, A., Sikora, M., Wróbel, Ł. RuleKit: A comprehensive suite for rule-based learning. *Knowledge-Based Systems*, 194:105480, 01 2020.
- [40] Guillet, F., Hamilton, H. *Quality Measures in Data Mining*, wolumen 43. 01 2007.
- [41] Gunning, D., Aha, D. DARPA’s explainable artificial intelligence (xai) program. *AI Magazine*, 40(2):44–58, Jun. 2019.
- [42] Hashemi, S., Shamsinjead Babaki, P. GA 2 RM: A GA-based action rule mining method. *International Journal of Computational Intelligence and Applications*, 20:2150012, 06 2021.
- [43] He, Z., Xu, X., Deng, S. Data mining for actionable knowledge: A survey. 02 2005.
- [44] He, Z., Xu, X., Deng, S., Ma, R. Mining action rules from scratch. *Expert Systems with Applications*, 29:691–699, 10 2005.
- [45] Henzel, J. Indukcja reguł akcji z danych zawierających informacje o przeżyciu. Praca magisterska, Politechnika Śląska, Gliwice, Polska, Czerw. 2019.
- [46] Im, S., Raś, Z. W. Action rule extraction from a decision table: Ared. An, A., Matwin, S., Raś, Z. W., Ślęzak, D., redaktorzy, *Foundations of Intelligent Systems*, strony 160–168, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.
- [47] Iman, R., Davenport, J. Approximations of the critical region of the friedman statistic. *Communications in Statistics-Theory and Methods*, 9:571–595, 01 1980.

- [48] Janssen, F., Fürnkranz, J. Heuristic rule-based regression via dynamic reduction to classification. *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two*, IJCAI'11, strona 1330–1335. AAAI Press, 2011.
- [49] Janssen, F., Fürnkranz, J. On the quest for optimal rule learning heuristics. *Machine Learning*, 78:343–379, 03 2010.
- [50] Kaplan, E. L., Meier, P. Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, 53(282):457–481, 1958.
- [51] Klösgen, W. *Explora: A Multipattern and Multistrategy Discovery Assistant*, strona 249–271. American Association for Artificial Intelligence, USA, 1996.
- [52] Kozielski, M., Skowron, A., Wróbel, Ł., Sikora, M. Regression rule learning for methane forecasting in coal mines. *Communications in Computer and Information Science*, 521:495–504, 05 2015.
- [53] Kryszkiewicz, M. Strong rules in large databases. *Traitement d'information et gestion d'incertitudes dans les systèmes à base de connaissances. Conférence internationale*, strony 1520–1527, 1998.
- [54] Letham, B., Rudin, C., McCormick, T., Madigan, D. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics*, 9:1350–1371, 09 2015.
- [55] Lin, X. MR-Apriori: Association rules algorithm based on MapReduce. *2014 IEEE 5th International Conference on Software Engineering and Service Science*, strony 141–144, 2014.
- [56] Liu, B., Abbass, H., McKay, R. Classification rule discovery with ant colony optimization. *IEEE Intelligent Informatics Bulletin*, 3:31–35, 01 2004.
- [57] Matyszok, P., Sikora, M., Wróbel, Ł. Covering approach to action rule learning. *Proceedings of 14th International Conference, BDAS 2018, Held at the 24th IFIP World Computer Congress, WCC 2018, Poznań, Poland*, strony 182–193, 09 2018.

- [58] Matyszok, P., Wróbel, Ł., Sikora, M. Bidirectional action rule learning. Cza-  
chórski, T., Gelenbe, E., Grochla, K., Lent, R., redaktorzy, *Computer and  
Information Sciences*, strony 220–228, Cham, 2018. Springer International  
Publishing.
- [59] Michalski, R. S. Discovering classification rules using variable-valued logic  
system VL1. *Proceedings of the 3rd International Joint Conference on Artificial  
Intelligence, IJCAI'73*, strony 162–172, San Francisco, CA, USA, 1973. Morgan  
Kaufmann Publishers Inc.
- [60] Mierswa, I., Klinkenberg, R. Rapidminer studio. <https://rapidminer.com/>,  
2021.
- [61] Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., Euler, T. Yale: Rapid pro-  
totyping for complex data mining tasks. *Proceedings of the 12th ACM SIGKDD  
International Conference on Knowledge Discovery and Data Mining, KDD  
'06*, strona 935–940, New York, NY, USA, 2006. Association for Computing  
Machinery.
- [62] Mitchell, T. M. *Machine Learning*. McGraw-Hill, New York, 1997.
- [63] Nemenyi, P. B. *Distribution-free multiple comparisons*. Praca doktorska,  
Princeton University, 1963.
- [64] Ossowski, S. *Sieci neuronowe w ujęciu algorytmicznym*. WNT, 1996.
- [65] Pao, Y., Takefuji, Y. Functional-link net computing: Theory, system architec-  
ture, and functionalities. *Computer*, 25:76 – 79, 06 1992.
- [66] Pawlak, Z. Rough sets. *International Journal of Computer & Information  
Sciences*, 11(5):341–356, Oct 1982.
- [67] Piatetsky-Shapiro, G., Matheus, C. J. The interestingness of deviations. *Pro-  
ceedings of the 3rd International Conference on Knowledge Discovery and Data  
Mining, AAAIWS'94*, strona 25–36. AAAI Press, 1994.
- [68] Pietraszek, T. Optimizing abstaining classifiers using ROC analysis. *ICML  
2005 - Proceedings of the 22nd International Conference on Machine Learning*,  
strony 665–672, 01 2005.

- [69] Powell, L., Gelich, A., Ras, Z. W. How to raise artwork prices using action rules, personalization and artwork visual features. *J. Intell. Inf. Syst.*, 57:583–599, 2021.
- [70] Quinlan, R. *C4.5: Programs for Machine Learning*, wolumen 1. Morgan Kaufmann Publishers, Inc., 1993.
- [71] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2020.
- [72] Ågotnes, T., Komorowski, J., Løken, T. Taming large rule models in rough set approaches. Żytkow, J. M., Rauch, J., redaktorzy, *Principles of Data Mining and Knowledge Discovery*, strony 193–203, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.
- [73] Raś, Z., Dardzińska, A. Action rules discovery based on tree classifiers and meta-actions. *ISMIS09*, wolumen 5722, strony 66–75, 09 2009.
- [74] Raś, Z., Dardzińska, A., Tsay, L.-S., Wasyluk, H. Association action rules. *Proceedings - IEEE International Conference on Data Mining Workshops, ICDM Workshops 2008*, strony 283 – 290, 01 2009.
- [75] Raś, Z. W., Wieczorkowska, A. Action-rules: How to increase profit of a company. Zighed, D. A., Komorowski, J., Żytkow, J., redaktorzy, *Principles of Data Mining and Knowledge Discovery*, strony 587–592, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.
- [76] Raś, Z. W., Wyrzykowska, E. Arogs: Action rules discovery based on grabbing strategy and lers. *Proceedings of 2007 ECML/PKDD Third International Workshop on Mining Complex Data*, strony 95–105. Citeseer, 2007.
- [77] Ribeiro, M. T., Singh, S., Guestrin, C. "why should i trust you?": Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, strona 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery.
- [78] Shamsinejadbabaki, P., Saraee, M., Blockeel, H. Causality-based cost-effective action mining. *Intelligent Data Analysis*, 17:1075–1091, 11 2013.



- [79] Shearer, C. The CRISP-DM model: the new blueprint for data mining. *Journal of data warehousing*, 5(4):13–22, 2000.
- [80] Sher, B.-Y., Shao, S.-C., Hsieh, W.-S. Mining regression rules and regression trees. Wu, X., Kotagiri, R., Korb, K. B., redaktorzy, *Research and Development in Knowledge Discovery and Data Mining*, strony 271–282, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.
- [81] Sikora, M. Filtracja zbioru reguł decyzyjnych wykorzystująca funkcje oceny jakości reguł. *Studia Informatica*, number 46, strony 57–72, 2001.
- [82] Sikora, M., Gudyś, A., Wróbel, Ł. GuideR: a guided separate-and-conquer rule learning in classification, regression, and survival settings. *Knowledge-Based Systems*, 173:1–14, 2019.
- [83] Sikora, M., Matyszok, P., Wróbel, Ł. SCARI: Separate and conquer algorithm for action rules and recommendations induction. *Information Sciences*, 2022.
- [84] Sikora, M., Simiński, K. Comparison of incomplete data handling techniques for neuro-fuzzy system. *Comput. Sci.*, 15:441–458, 2014.
- [85] Sikora, M., Wróbel, Ł. Data-driven adaptive selection of rule quality measures for improving rule induction and filtration algorithms. *International Journal of General Systems*, 42, 08 2013.
- [86] Silberschatz, A., Tuzhilin, A. What makes patterns interesting in knowledge discovery systems. *IEEE Trans. Knowl. Data Eng.*, 8:970–974, 01 1996.
- [87] Stefanowski, J. Handling continuous attributes in discovery of strong decision rules. Polkowski, L., Skowron, A., redaktorzy, *Rough Sets and Current Trends in Computing*, strony 394–401, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.
- [88] Stefanowski, J. Rough set based rule induction techniques for classification problems. *6th European Congress of Intelligent Techniques and Soft Computing*, strony 107–119, 1998.
- [89] Stefanowski, J., Vanderpooten, D. Induction of decision rules in classification and discovery-oriented perspectives. *International Journal of Intelligent Systems*, 16, 01 2001.

- [90] Studer, S., Bui, B., Drescher, C., Hanuschkin, A., Winkler, L., Peters, S., Müller, K.-R. Towards CRISP-ML(Q): A machine learning process model with quality assurance methodology. *Machine Learning and Knowledge Extraction*, 3:392–413, 04 2021.
- [91] Stąpor, K., Ksieniewicz, P., García, S., Woźniak, M. How to design the fair experimental classifier evaluation. *Appl. Soft Comput.*, 104:107219, 2021.
- [92] Tarnowska, K., Raś, Z. Sentiment analysis of customer data. *Web Intelligence*, 17:1–21, 11 2019.
- [93] Tarnowska, K., Raś, Z. NLP-based customer loyalty improvement recommender system (CLIRS2). *Big Data and Cognitive Computing*, 5:4, 01 2021.
- [94] Tarnowska, K., Raś, Z., Daniel, L. *Recommender System for Improving Customer Loyalty*. Springer International Publishing, 01 2020.
- [95] Touati, H., Raś, Z. W., Studnicki, J., Wieczorkowska, A. A. Mining surgical meta-actions effects with variable diagnoses' number. Andreasen, T., Christiansen, H., Cubero, J.-C., Raś, Z. W., redaktorzy, *Foundations of Intelligent Systems*, strony 254–263, Cham, 2014. Springer International Publishing.
- [96] Trépos, R., Salleb, A., Cordier, M.-O., Masson, V., Gascuel-Odoux, C. Building actions from classification rules. *Knowledge and Information Systems*, 34, 02 2012.
- [97] Tuzhilin, A., Wang, K., Jiang, Y. Mining actionable patterns by role models. *22nd International Conference on Data Engineering*, strona 16, Los Alamitos, CA, USA, apr 2006. IEEE Computer Society.
- [98] Tzacheva, A., Bagavathi, A., Ganesan, P. MR-random forest algorithm for distributed action rules discovery. *Data Mining and Knowledge Discovery*, 6:15–30, 09 2016.
- [99] Tzacheva, A., Sunny, M., Mummoju, P. MR-Apriori count distribution algorithm for parallel Action Rules discovery. *2016 IEEE International Conference on Knowledge Engineering and Applications (ICKEA)*, strony 127–132, 09 2016.
- [100] Weiss, S., Indurkha, N. Rule-based machine learning methods for functional prediction. *Journal of Artificial Intelligence Research*, 3, 11 1995.

- [101] Wilcoxon, F. Individual comparisons by ranking methods. *Biometrics*, 1:196–202, 1945.
- [102] Wohlrab, L., Fürnkranz, J. A review and comparison of strategies for handling missing values in separate-and-conquer rule learning. *Journal of Intelligent Information Systems*, 36:73–98, 02 2011.
- [103] Wróbel, Ł., Sikora, M. Censoring weighted separate-and-conquer rule induction from survival data. *Methods of information in medicine*, 53, 02 2014.
- [104] Wróbel, Ł., Sikora, M., Michalak, M. Rule quality measures settings in classification, regression and survival rule induction — an empirical approach. *Fundamenta Informaticae*, 149:419–449, 12 2016.
- [105] Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Yu, P. S., Zhou, Z.-H., Steinbach, M., Hand, D. J., Steinberg, D. Top 10 algorithms in data mining. *Knowl. Inf. Syst.*, 14(1):1–37, dec 2007.
- [106] Yang, H., Rudin, C., Seltzer, M. Scalable bayesian rule lists. *Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML'17*, strona 3921–3930. JMLR.org, 2017.
- [107] Yang, Q., Yin, J., Ling, C., Pan, R. Extracting actionable knowledge from decision trees. *IEEE Transactions on Knowledge and Data Engineering*, 19(1):43–56, 2007.
- [108] Yao, Y. Y., Zhong, N. An analysis of quantitative measures associated with rules. Zhong, N., Zhou, L., redaktorzy, *Methodologies for Knowledge Discovery and Data Mining*, strony 479–488, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.
- [109] Zomorodi, M., Abdar, M., Davarzani, Z., Zhou, X., Pławiak, P., Acharya, U. R. Hybrid particle swarm optimization for rule discovery in the diagnosis of coronary artery disease. *Expert Systems*, 38:1–17, 01 2021.
- [110] Šimůnek, M. Academic KDD project LISp-miner. *Intelligent Systems Design and Applications*, strony 263–272. Springer, 01 2003.

*Bibliografia*

- [111] Żabiński, K., Zielosko, B. Decision rules construction: Algorithm based on EAV model. *Entropy*, 23:14, 12 2020.