

System indukcji reguł gradualnych z danych
porządkowych o powiązaniach monotonicznych
między częścią warunkową a decyzyjną

Wojciech Kotłowski

8 września 2004

Spis treści

1	Wstęp	3
2	Elementy teorii zbiorów rozmytych i przybliżonych	8
2.1	Logika rozmyta	8
2.1.1	Zbiór rozmyty	8
2.1.2	Operacje na zbiorach rozmytych	9
2.1.3	Relacje rozmyte	9
2.1.4	Implikacja, reguły i wnioskowanie	10
2.2	Klasyczna teoria zbiorów przybliżonych	11
2.2.1	Tablica informacyjna i tablica decyzyjna	11
2.2.2	Relacja nierozróżnialności, górne i dolne przybliżenie	11
2.2.3	Jakość klasyfikacji, redukt	12
2.2.4	Reguły decyzyjne	13
2.3	Podejście do zbiorów przybliżonych oparte na relacji dominacji	14
2.3.1	Relacja dominacji, dolne i górne przybliżenie	14
2.3.2	Jakość klasyfikacji, redukt	15
2.3.3	Reguły decyzyjne	15
3	Reguły gradualne	17
3.1	Wprowadzenie	17
3.1.1	Reguły decyzyjne z monotoniczną zależnością między przesłanką a konkluzją	18
3.1.2	Rozszerzenie na reguły z wieloma przesłankami	19
3.2	Podejście rozmyto-przybliżone do indukcji reguł	19
3.2.1	Przypadek dla reguł z jedną przesłanką	20
3.2.2	Przypadek dla reguł z wieloma przesłankami	22
3.2.3	Zasada maksymalnej ostrożności	23
3.2.4	Wyznaczanie wiarygodności przynależności obiektów do dolnych i górnych przybliżeń	24
3.3	Wnioskowanie rozmyto-przybliżone	24
3.3.1	Wnioskowanie w oparciu o ścisłą przesłankę	25
3.3.2	Wnioskowanie w oparciu o rozmytą przesłankę	26
3.4	Przykładowy zbiór danych	27

4	Indukcja reguł	31
4.1	Wprowadzenie	31
4.2	Uogólnianie poprzez redukcję atrybutów	32
4.2.1	Jakość klasyfikacji i redukty	32
4.2.2	Redukcja atrybutów	33
4.3	Indukcja reguł decyzyjnych z macierzy dominacji	34
4.3.1	Generowanie reduktów lokalnych	35
4.3.2	Tworzenie reguł	36
4.3.3	Algorytm indukcji	37
4.3.4	Selekcja reguł na podstawie wsparcia	39
4.3.5	Selekcja reguł na podstawie stopnia przekrywania	41
4.3.6	Przykład indukcji reguł	42
5	Opis zbioru danych w oparciu o skalę ilorazową	47
5.1	Uwagi wstępne	47
5.2	Minimalizacja zbioru reguł w oparciu o odległości między obiektami	48
5.2.1	Miara jakości reguł na podstawie odległości	48
5.2.2	Selekcja reguł na podstawie odległości	50
5.2.3	Przykład selekcji reguł	52
5.3	Problem regresji wielowymiarowej za pomocą funkcji odcinkami liniowej	54
5.3.1	Programowanie matematyczne	54
5.3.2	Przypadek regresji przy zadanych punktach charakterystycznych funkcji	57
5.3.3	Przypadek regresji dla punktów charakterystycznych funkcji traktowanych jako zmienne decyzyjne	60
5.3.4	Przykład regresji	65
5.3.5	Konstrukcja reguł na podstawie modelu funkcji odcinkami liniowej	69
6	Podsumowanie	73
	Bibliografia	76
	Dodatek: Opis programu i instrukcja użytkownika	79
6.1	Opis ogólny	79
6.2	Funkcje programu	81
6.2.1	Edycja tablicy decyzyjnej	81
6.2.2	Konstrukcja kryteriów	81
6.2.3	Przekształcanie atrybutów, dolne i górne przybliżenia	82
6.2.4	Generowanie i przeglądanie reguł decyzyjnych	82
6.2.5	Regresja	83
6.2.6	Prezentacja graficzna wyników	84
6.2.7	Klasyfikacja	84

Rozdział 1

Wstęp

W ostatnich dekadach, w jednym z podstawowych nurtów informatyki, sztucznej inteligencji, dało się zauważyć dążenie ku systemom uczącym się, nie wymagającym wprowadzania dużych ilości wiedzy. Systemy takie powinny na podstawie dostarczanych im danych same odkryć, jaki jest najlepszy sposób dotarcia do wzorców ukrytych w danych, czyli do wiedzy. Wyodrębniła się nawet poddziedzina sztucznej inteligencji, zwana *maszynowym uczeniem się*, zajmująca się rozwojem technik uczenia się i budową systemów uczących się [23].

W ramach teorii uczenia się, wyjątkowo ważną rolę pełni *uczenie indukcyjne*. Polega ono na uogólnieniu otrzymanych danych wejściowych poprzez wygenerowanie z nich i odpowiednie przedstawienie pewnej wiedzy, nazywanej *hipotezą*. Wiedza ta umożliwia wyjaśnienie danych początkowych oraz predykcję przyszłych danych. Do tworzenia i opisu hipotez wykorzystuje się tzw. *pojęcia*, czyli zbiory obiektów przypisanych z powodu pewnych cech wspólnych do tej samej klasy. Przynależność do pojęcia wykazuje cechy zwrotności, symetrii i przechodności, z matematycznego punktu widzenia pojęcia są więc klasami abstrakcji relacji równoważności, dokonującej podziału zbioru obiektów. Bardzo często termin *pojęcie* zastępowany jest terminem *klasa*, co prowadzi do nazwania powyższych zagadnień problemami *klasyfikacji*.

Pierwszym problemem, na jaki napotykamy w teorii uczenia się indukcyjnego, jest znalezienie odpowiedniego sposobu reprezentacji wygenerowanych hipotez. Zaproponowano bardzo wiele podejść do tego zagadnienia, używając struktur takich jak drzewa decyzyjne, reguły, sieci bayesowskie, skupienia, predykaty logiczne a nawet wagi na połączeniach między sztucznymi neuronami [3]. Wymienione sposoby reprezentacji można oceniać rozważając przynajmniej dwa kryteria. Jednym z nich będzie z pewnością trafność klasyfikacji przy stosowaniu utworzonej już hipotezy do obiektów, o których nie wiemy, do jakiego pojęcia (klasy) powinny należeć. Duży wpływ będzie miał tutaj zakres hipotez, możliwych do wygenerowania za pomocą rozważanych struktur, czyli kwestia, jaką część wszystkich możliwych hipotez stanowią hipotezy, które można zapisać w języku danej reprezentacji (problem ten nazywa się zazwyczaj *obciążeniem indukcyjnym*).

Prócz zdolności predykcyjnych systemu w grę wchodzi jeszcze jedno kryterium, często mające nawet większe znaczenie — wyjaśnienie zbioru danych, i to w możliwie najprostszy, zrozumiały dla człowieka sposób. Niestety, jak piszą Dubois, Nguyen, Prade i Sugeno [8], te dwa kryteria są zazwyczaj wzajemnie przeciwstawne, gdyż im bardziej dokładny i precyzyjny jest model, tym bardziej skomplikowany i trudniejszy jest on do wyjaśnienia. Dobrym przykładem są tutaj sieci neuronowe, cechujące się wysokimi zdolnościami predykcyjnymi, ale hipoteza przez nie reprezentowana jest bardzo trudna do wyjaśnienia. Podejście, którego głównym celem jest stworzenie jak najdokładniej klasyfikującej hipotezy, zwykle jest silniej związane z metodami numerycznymi i modelowaniem za pomocą skomplikowanych funkcji, podczas gdy sam system działa na zasadzie „czarnej skrzynki” — ważne jest wyłącznie to, aby dawał poprawne wyniki, natomiast nikt nie „zagląda” do jego wnętrza i nie próbuje zrozumieć jego budowy. W przypadku drugiego podejścia, dla którego ważniejsze jest wyjaśnienie danych, stosowane są metody symboliczne, a hipotezy proste są preferowane nad skomplikowane. W typowych problemach pojawia się natomiast przetarg między oboma kryteriami.

Sposobem reprezentacji hipotez, który z jednej strony pozwala na klasyfikację z wysoką skutecznością, a zarazem jest niezwykle prosty i czytelny w opisie, są reguły decyzyjne. Pod tą nazwą kryją się wyrażenia postaci: „JEŚLI warunki (przesłanki) TO decyzja (konkluzja)”, posiadające bardzo intuicyjne dla człowieka znaczenie. Podejście regułowe ma swoje korzenie w rachunku predykatów i logice formalnej, a same reguły są zazwyczaj rodzajem implikacji. Część warunkowa reguły (przesłanka) składa się najczęściej z koniunkcji pojedynczych warunków, zwanych elementarnymi (selektorów). Część decyzyjna, przynajmniej w przypadku reguł rozważanych w obrębie tej pracy, ma postać wyrażenia pozwalającego na klasyfikację spełniającego przesłankę obiektu. Wyrażeniem takim jest zazwyczaj przypisanie obiektów do pewnej klasy (pojęcia) [31].

Podstawowym zadaniem teorii uczenia indukcyjnego jest szukanie jak najlepszych sposobów generowania hipotez z danych, co w przypadku podejścia regułowego nazywane jest zwykle *indukcją reguł*. Dodatkowo, sam zbiór reguł może mieć różną postać — może to być zbiór wyczerpujący, zawierający wszystkie możliwe do wygenerowania *reguły minimalne*, czyli takie, że biorąc jakąkolwiek parę spośród nich, żadna z reguł nie będzie od drugiej silniejsza (reguła silniejsza to reguła o słabszej przesłance i/lub silniejszej konkluzji). Inna możliwość to wygenerowanie zbioru minimalnego reguł minimalnych, czyli takiego, który właściwie klasyfikuje zbiór danych uczących, a zarazem jakikolwiek jego podzbiór właściwy nie posiada już tej własności. Niestety, udowodniono (poprzez transformację od problemu minimalnego pokrycia), że problem znalezienia minimalnego zbioru reguł jest NP-zupełny. Wynika stąd, że nie znamy (i najprawdopodobniej nie istnieje) algorytmu wielomianowego, który rozwiązywałby ten problem. Musimy posługiwać się heurystykami lub algorytmami dokładnymi, ale działającymi stosunkowo szybko dla rzeczywistych instancji.

Ponieważ nie ma szybkiego algorytmu dokładnego, zaproponowano wiele heurystycznych algorytmów indukcji reguł, z których chronologicznie pierwsze

były AQ oraz CN2. Spośród innych algorytmów można wymienić również LEM2 oraz jego ulepszoną wersję MODLEM. W niniejszej pracy jedynym rozważanym algorytmem będzie, oparte o tzw. macierz dominacji, znajdowanie reduktów lokalnych i generacja na ich podstawie reguł bazujących na obiektach [6].

W celu zwiększenia siły wyrazu podejścia regułowego (i nie tylko regułowego) w praktycznych zastosowaniach szukano sposobu radzenia sobie z niepewnościami tkwiącymi w rzeczywistych danych. Są dwie znane formy rozwiązywania problemów z tego typu niepewnościami — pomoc nadeszła ze strony szybko rozwijających się nowych dziedzin: logiki rozmytej oraz teorii zbiorów przybliżonych. Zbiory rozmyte, wprowadzone przez Zadeha [38], pozwalają radzić sobie z niedokładnie zdefiniowanymi stanami, niejasno określonymi pojęciami, brakiem precyzji w opisie preferencji. Z kolei zbiory przybliżone, zaproponowane przez Pawłaka [25], pomagają w analizie danych w przypadku niespójności i dwuznaczności. Obie teorie wniosły w ciągu ostatnich lat duży wkład w rozwój dziedziny uczenia indukcyjnego.

Problem rozpatrywany w tej pracy dotyczy zastosowania reguł w systemach wielokryterialnego wspomaganie decyzji. Jest to klasa zagadnień, dla których na podstawie informacji preferencyjnych decydenta, zbioru wariantów i spójnej rodziny kryteriów, rozwiązuje się problem należący do jednej z trzech następujących kategorii: klasyfikacji (sortowania), wyboru lub porządkowania [27]. Klasyfikacja to podział całego zbioru wariantów na podzbiory, uszeregowane pod względem preferencji (na zbiór klas narzucona jest relacja porządku zupełnego). Problem wyboru to wyznaczenie podzbioru najlepszych wariantów. Porządkowanie to znajdowanie relacji pre-porządku, częściowego lub zupełnego w zbiorze wariantów. Podstawową zmianą w przypadku stosowania reguł do wspomaganie decyzji, jest pojawienie się kryteriów, czyli atrybutów ze skalą preferencji. Tym samym należy zrewidować dotychczasowy sposób reprezentacji i konstrukcji reguł, przede wszystkim dlatego, że pojawiają się niespójności nowego typu — związane z wzajemnym dominowaniem się wariantów, a nie, jak poprzednio, wyłącznie z ich nierozróżnialnością (tymi samymi wartościami na atrybutach). Inny rodzaj niespójności pociągnął za sobą konieczność wprowadzenia modyfikacji do teorii zbiorów przybliżonych, co zaowocowało stworzeniem podejścia do zbiorów przybliżonych opartego na relacji dominacji (ang. *Dominance Based Rough Set Approach — DRSA*) [18]. Pozwoliło to na zastosowanie algorytmów indukcji reguł dla danych z relacją porządku na dziedzinach atrybutów do rozwiązywania problemów klasyfikacji, a później również porządkowania.

Naturalnym rozwinięciem tego podejścia, wraz z zastosowaniem w nim zbiorów rozmytych, okazała się reprezentacja modelu preferencji w postaci *reguł gradualnych*. Reguły takie mają następującą składnię: „JEŚLI wiarygodność warunku jest większa od α , TO wiarygodność decyzji jest większa od β ”; przez *wiarygodność* rozumiemy tutaj stopień przynależności do odpowiednich zbiorów rozmytych. Wykorzystując wyłącznie zależności monotoniczne, nie biorąc pod uwagę wartości liczbowych i odległości na skali, w oparciu o przekształcenia zachowujące porządek, zaproponowano [14] nowy sposób wyznaczania górnych i dolnych przybliżeń oraz rozmytego wnioskowania. Wszystkie działania przeprowadzane na zbiorze danych mają miejsce dopiero po uprzednim przekształceniu,

dla wszystkich obiektów z takiego zbioru, wartości atrybutów na stopnie przynależności do zbiorów rozmytych określających przynależność do pojęć (zmiennych lingwistycznych) utworzonych na atrybutach. Mając semantyczną wiedzę o zmiennych lingwistycznych i znając monotoniczne związki wiążące je z wiarygodnością decyzji, można utożsamić te zmienne z kryterium (pojawia się skala preferencji).

Niniejsza praca dotyczy właśnie podejścia do klasyfikacji opartego na regułach gradualnych. Jedną z podstawowych różnic między starszymi modelami, a modelem reguł gradualnych, jest fakt, że obiekty na wszystkich kryteriach, łącznie z decyzją, przyjmują (po przekształceniu) wartości ciągłe. Stąd można się spodziewać, że obiekty o tych samych wartościach decyzji będą występowały bardzo rzadko lub nawet wcale. Powoduje to trudności w regułowym opisie zbioru, który na skutek gęstej klasyfikacji staje się obszerniejszy i trudniejszy i do interpretacji. Cele pracy, jakie zostały wytyczone podczas analizy powyższych problemów, przedstawiają się następująco:

1. Zapoznanie się i przeanalizowanie podejścia opartego na regułach gradualnych. Dodatkowo, ponieważ nie istniało wcześniej oprogramowanie implementujące te metody, napisanie systemu indukcji reguł, posiadającego podstawową funkcjonalność.
2. Indukcja reguł — na podstawie operacji na górnych i dolnych przybliżeniach można stworzyć funkcje (reguły) gradualne, które pozwalają klasyfikować nowe obiekty poprzez wskazanie minimalnej i maksymalnej wartości ich decyzji. Funkcje takie są jednak jeszcze zbyt mało ogólne, ponieważ wyciągają możliwie najostrożniejsze wnioski. Pojawia się więc pomysł indukcji reguł bardziej ogólnych, lepiej wyjaśniających tablicę decyzyjną.
3. Klasyfikacja przy wykorzystaniu odległości między obiektami — jeśli zmodyfikowalibyśmy założenia dotyczące własności skali stosowanej na kryteriach przez umożliwienie wiarygodnego wypowiedzenia się na temat porównań odległości między obiektami na wszystkich kryteriach i decyzji, otworzyłyby to ogromne możliwości zastosowania różnego rodzaju metod potrzebujących własności numerycznych, począwszy od dyskretyzacji, analizy skupień obiektów, na regresji skończywszy. Wybrane zostały dwa podejścia: opis wciąż za pomocą reguł, ale tym razem z możliwością dodatkowej redukcji zbioru reguł poprzez odpowiednik dyskretyzacji, oraz regresja za pomocą funkcji odcinkami liniowej.

W rozdziale drugim przedstawione zostały zbiory elementy teorii zbiorów rozmytych oraz teorii zbiorów przybliżonych, przy czym przy tych ostatnich rozpatrzono dwa podejścia — klasyczne oraz oparte na relacji dominacji. Trzeci rozdział poświęcono opisowi i analizie reguł gradualnych. W rozdziale czwartym zamieszczony został opis indukcji reguł z funkcji gradualnych za pomocą generowania reduktów lokalnych; pokazano również oryginalny sposób usuwania nadmiarowych reguł z wyczerpującego ich zbioru. Piąty rozdział dotyczy metod opartych na własnościach numerycznych — z możliwością porównywania

odległości. Wprowadzono nowy sposób redukcji wyczerpującego zbioru reguł, ale większą część rozdziału zajmuje dokładny opis dwóch metod regresji, pozwalający dodatkowo utworzyć w przestrzeni kryteriów siatkę reguł z konkluzją w postaci funkcji liniowej. Ostatni z rozdziałów stanowi zakończenie i podsumowanie pracy.

Rozdział 2

Elementy teorii zbiorów rozmytych i przybliżonych

2.1 Logika rozmyta

Logika rozmyta (ang. *fuzzy logic*) wprowadzona została w latach sześćdziesiątych przez L. Zadeha [38], w celu zastosowania w teorii sterowania, mającej trudności z rozwiązywaniem rzeczywistych problemów, z reguły silnie nieliniowych i opierających się próbom stworzenia nawet przybliżonych modeli. Podstawowym pomysłem, odróżniającym nowe podejście od klasycznego, było wykorzystanie wiedzy ekspertów, próba pozyskania informacji od ludzi, którzy zdobyli ją poprzez doświadczenie i często posługiwali się nią intuicyjnie. Zamiast, jak dotychczas, budować jak najdokładniejszy model kontrolowanego procesu (zwykle układ równań różniczkowych, linearyzowany w celu umożliwienia rozwiązania), układano zwykle, lingwistyczne reguły, opisujące sposób wnioskowania eksperta o systemie. Następnie, poprzez zastosowanie funkcji przynależności do zbiorów, przy przeprowadzaniu wnioskowania otrzymywane były nie tylko lingwistyczne stwierdzenia, lecz również kryjące się za nimi wartości. W ten sposób udało się połączyć dwa przeciwstawne kierunki rozwiązywania problemów: modelowanie (zbudowanie funkcji jak najdokładniej symulującej działanie opisywanego procesu) oraz wyjaśnianie (zbudowanie systemu pozwalającego odkryć sposób działania, wiedzę zrozumiałą dla ludzi) [8].

2.1.1 Zbiór rozmyty

Logika rozmyta opiera się na pojęciu *zbioru rozmytego*, czyli zbioru danych A wraz z funkcją $\mu_A: X \rightarrow [0, 1]$, nazywaną *funkcją przynależności*, gdyż każdemu elementowi z przestrzeni informacji X przyporządkowuje liczbę rzeczywistą, określającą w jakim stopniu dany element należy do zbioru A [36]. Funkcja przynależności może mieć dowolny przebieg, jednak z reguły wymagane jest, aby jej wysokość (największa wartości na zbiorze X) była równa 1. Z mate-

matycznego punktu widzenia następuje tu rozszerzenie logiki dwuwartościowej (w której ocena prawdziwości każdego stwierdzenia może przyjąć dwie możliwe wartości: *prawdę* lub *falsz*) do, wymyślonej przez J. Łukasiewicza już w latach dwudziestych, logiki wielowartościowej (ocena prawdziwości jest liczbą, może przyjąć continuum wartości z przedziału $[0,1]$).

2.1.2 Operacje na zbiorach rozmytych

Następnym krokiem jest przededefiniowanie operacji na zbiorach, aby możliwe było zastosowanie ich do zbiorów rozmytych, przy równoczesnym zachowaniu wszystkich klasycznych własności [22]. W naturalny sposób wprowadza się funkcję przynależności do $\neg A$ (negacji zbioru), definiowaną jako:

$$\mu_{\neg A}(x) = 1 - \mu_A(x) \quad (2.1)$$

Niejednoznaczny jest natomiast sposób konstrukcji funkcji przynależności dla sumy i iloczynu (części wspólnej) zbiorów. Z jednej strony mogą mieć one postać:

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)) \quad (2.2)$$

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)) \quad (2.3)$$

Jednak równie uzasadniona jest także odmienna definicja:

$$\mu_{A \cup B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x)\mu_B(x) \quad (2.4)$$

$$\mu_{A \cap B}(x) = \mu_A(x)\mu_B(x) \quad (2.5)$$

Można łatwo pokazać, że wszystkie wymienione funkcje spełniają własności klasycznej sumy i iloczynu zbiorów, a ich wartości zawierają się zawsze w przedziale $[0,1]$. Definicje 2.4 oraz 2.5 wprowadzają funkcje bardziej czułe na zmiany oraz szacujące ostrożniej. Niestety, żadne z powyższych funkcji nie zachowują porządku elementów, co dyskwalifikuje je w przypadku posługiwania się wyłącznie zależnościami porządkowymi (szerzej w rozdziale 3). Operacja iloczynu zwana jest zwykle *t-normą*, a operacja sumy — *s-normą*.

2.1.3 Relacje rozmyte

W podobny sposób możemy potraktować klasyczne relacje (będące, z punktu widzenia teorii mnogości, podzbiorami iloczynu kartezjańskiego), przy czym można do nich stosować wyżej opisane operacje na zbiorach. Dodatkową operację, złożenie dwóch relacji $R \circ S$, gdzie $R \subset X \times Y$, $S \subset Y \times Z$, definiujemy w następujący sposób:

$$\mu_{R \circ S}(x, z) = \max_{y \in Y} (\min(\mu_R(x, y), \mu_S(y, z))) \quad (2.6)$$

przy złożeniu takie nazywane jest złożeniem typu *max-min*. Innym rodzajem złożenia jest *max-product*, $R \bullet S$:

$$\mu_{R \circ S}(x, z) = \max_{y \in Y} (\min(\mu_R(x, y) \cdot \mu_S(y, z))) \quad (2.7)$$

2.1.4 Implikacja, reguły i wnioskowanie

Implikację $A \rightarrow B$ można zastąpić równoważnym logicznie wyrażeniem $\neg A \cup B$, a następnie wyrazić przez relację $R = A \times B \cap \neg A \times B$, co pozwala zapisać funkcję przynależności, jako:

$$\mu_{A \rightarrow B}(x, y) = \max(\min(\mu_A(x), \mu_B(y)), 1 - \mu_A(x)) \quad (2.8)$$

co określane jest jako *implikacja Zadeha*. Często spotykana jest również *implikacja Mamdamiego*:

$$\mu_{A \rightarrow B}(x, y) = \min(\mu_A(x), \mu_B(y)) \quad (2.9)$$

Termy logiczne, w klasycznej logice przyjmujące wyłącznie dwie wartości (prawdę bądź fałsz), mogą zostać skojarzone ze zbiorami rozmytymi w taki sposób, że stopień prawdziwości termu odpowiada funkcji przynależności do zbioru skojarzonego z takim termem. Wykorzystując to, możemy przystąpić do tworzenia reprezentacji reguł. Część warunkowa reguły składa się z koniunkcji (iloczynu) warunków elementarnych, które mogą być reprezentowane jako termy logiczne wyrażone poprzez zbiory rozmyte (prawdziwe dla elementów, które należą do tych zbiorów). Samą regułę można przedstawić w postaci implikacji. Stąd stwierdzeniu, że reguła jest spełniona (które traktujemy jako zbiór rozmyty, do którego należą elementy spełniające regułę), możemy przypisać funkcję przynależności, konstruowaną na wyżej opisanych zasadach.

Teraz łatwo można wprowadzić do formalizmu rozmytego wnioskowanie dedukcyjne (wyciąganie wniosków wynikających z przesłanek na podstawie implikacji), zarówno postaci *modus ponens*, $A \wedge (A \rightarrow B) \rightarrow B$, jak i *modus tolens*, $\neg B \wedge (A \rightarrow B) \rightarrow \neg A$. Dokonujemy tego poprzez złożenie relacji — przykładowo, jeśli przesłanka określona jest przez funkcję przynależności do pewnego zbioru rozmytego A' , a regułę traktujemy jako relację R , to konkluzja B' również jest zbiorem rozmytym, przy czym $B' = A' \circ R$. Dzięki temu nie tylko reguły mają postać rozmytą, lecz również w naturalny sposób wprowadza się rozmyte przesłanki. Tworzy to bardzo dobre narzędzie do modelowania, nawet silnie nieliniowych, systemów [35, 37].

Logika rozmyta, dzięki połączeniu w naturalny i intuicyjny sposób świata numerycznego z symbolicznym, zdobyła dużą popularność i licznych zwolenników, a także doczekała się zastosowania w rozwiązywaniu rzeczywistych problemów sterowania. Zdobyła sobie również silną pozycję w badaniach teoretycznych dotyczących szeroko pojętej sztucznej inteligencji.

2.2 Klasyczna teoria zbiorów przybliżonych

Teoria zbiorów przybliżonych, zaproponowana na początku lat osiemdziesiątych przez Z. Pawłaka [25], jest narzędziem matematycznym pozwalającym radzić sobie ze zbiorami danych zawierającymi niespójności. Wynikają one ze skończonej rozdzielczości języka opisu danych z pewnego zbioru uniwersalnego U . Dokładniej, w rzeczywistych przypadkach mamy do czynienia z opisem za pomocą skończonej ilości atrybutów, z których każdy ma dziedzinę o skończonej mocy (przestaje być to prawdą w przypadku atrybutów o wartościach liczbowych, gdzie dziedzina może być nieskończona lub nawet nieprzeliczalna, ale istnieją metody przekształcenia takiego zbioru na zbiór o skończonej liczbie elementów — tzw. dyskretyzacja). Mówimy, że wiedza ma strukturę ziarnistą (granularną), gdyż na skutek opisu za pomocą skończonej ilości informacji, relacja nierozróżnialności (do której należą pary obiektów opisane tymi samymi wartościami na wybranych atrybutach) dokonuje podziału (partycji) zbioru danych U na klasy abstrakcji, zwane zbiorami elementarnymi, w obrębie których elementy są nierozróżnialne, gdyż ich opis za pomocą dostępnej wiedzy jest taki sam.

W przypadku, gdy wyodrębnimy ze zbioru atrybutów A dwa podzbiory — atrybutów decyzyjnych D oraz atrybutów warunkowych C , zbiór obiektów opisany w ten sposób nazywany jest tablicą decyzyjną. Można dokonać za pomocą obu zbiorów podziału U na klasy abstrakcji (zbiory C -elementarne i D -elementarne). Jeśli każdy ze zbiorów C -elementarnych zawiera się w jakimś zbiorze D -elementarnym, mamy do czynienia ze zbiorem spójnym, w przeciwnym przypadku pojawia się niespójność, którą można interpretować w następujący sposób: zbiór wartości na atrybutach warunkowych C nie wystarcza do jednoznacznego ustalenia wartości na zbiorze atrybutów decyzyjnych D , tzn. nie istnieje funkcja $f: C \rightarrow D$ opisująca zbiór obiektów z tablicy decyzyjnej.

2.2.1 Tablica informacyjna i tablica decyzyjna

Z formalnego punktu widzenia, definiujemy *tablicę informacyjną* jako uporządkowaną czwórkę $IT = \langle U, Q, V, f \rangle$, przy czym $U, Q, V \neq \emptyset$ oraz skończone, gdzie U jest zbiorem obiektów, Q — zbiorem atrybutów, V — zbiorem wartości atrybutów, $V = \bigcup_{q \in Q} V_q$, V_q jest dziedziną atrybutu q , natomiast $f: U \times Q \rightarrow V$ jest funkcją zupełną taką, że $f(x, q) \in V_q$ dla każdego $q \in Q$, $x \in U$, zwaną funkcją informacyjną [30].

Jeśli dokonamy podziału zbioru atrybutów Q na dwa podzbiory — atrybutów warunkowych C i decyzyjnych D , $Q = C \cup D$, $C \cap D = \emptyset$, tablicę taką nazywamy *decyzyjną* i oznaczana przez DT . Stosunkowo często D jest zbiorem jednoelementowym, tzn. $D = d$ [33].

2.2.2 Relacja nierozróżnialności, górne i dolne przybliżenie

Zdefiniujmy relację binarną I_B w następujący sposób [26]:

$$I_B = \{(x, y) : \forall q \in B \ f(x, q) = f(y, q)\} \quad (2.10)$$

Relację tą nazwiemy *relacją nierozróżnialności* (ang. *indiscernibility relation*), gdyż należą do niej pary elementów, które nie różnią się na żadnym atrybucie ze zbioru B . Jest zwrotna, symetryczna i przechodnia, jest więc relacją równoważności, a tym samym indukuje podział (partycję) zbioru U , oznaczany U/I_B , przy czym klasy abstrakcji nazywane są *zbiorami B -elementarnymi* i oznaczane $B(x)$.

B -dolne i B -górne przybliżenie pewnego zbioru $X \subset U$, oznaczamy odpowiednio przez $\underline{B}(X)$, $\overline{B}(X)$ i definiujemy w sposób następujący:

$$\underline{B}(X) = \{x \in U : B(x) \subseteq X\} \quad (2.11)$$

$$\overline{B}(X) = \{x \in U : B(x) \cap X \neq \emptyset\} \quad (2.12)$$

Interpretacja powyższych zbiorów jest następująca: do górnego przybliżenia należą wszystkie obiekty, należące do zbiorów elementarnych, w obrębie których występuje przynajmniej jeden element ze zbioru X (stąd, skoro są z nim nierozróżnialne, istnieje przesłanka, świadcząca o możliwości przynależności takich obiektów do X). Z kolei dolne przybliżenie składa się z obiektów, których całe klasy abstrakcji zawierają się w X (co pozwala nam stwierdzić, że z pewnością należą do X). Zachodzi $\underline{B}(X) \subseteq X \subseteq \overline{B}(X)$, a różnicę między B -górnym i B -dolnym przybliżeniem nazywamy *B -brzegiem* i oznaczamy przez $BN_B(X)$. Jeśli brzeg jest pusty, zbiór X nazywamy spójnym, w przeciwnym przypadku X jest zbiorem przybliżonym.

W szczególności, mając tablicę decyzyjną, rozpatrujemy podział U/I_C oraz $U/I_D = \{Cl_1, Cl_2, \dots, Cl_n\}$ i wyznaczamy wartości C -górných i C -dolnych przybliżeń dla wszystkich zbiorów Cl_i , tzn. $\underline{C}(Cl_i)$ oraz $\overline{C}(Cl_i)$

2.2.3 Jakość klasyfikacji, redukt

Jakość klasyfikacji definiujemy jako:

$$\gamma(P, R) = \frac{\sum_{X \in U/I_R} |\underline{P}(X)|}{|U|} \quad (2.13)$$

Powyższy współczynnik określa zdolność opisu przez zbiór P podziału indukowanego przez zbiór R . Zazwyczaj przyjmujemy $R = D$ oraz $P \subseteq C$. Mówimy, że tablica decyzyjna DT jest spójna, gdy $\gamma(C, D) = 1$.

Reduktem dla tablicy decyzyjnej nazywamy każdy zbiór $P \subseteq C$ spełniający następujące warunki:

$$\gamma(P, D) = \gamma(C, D) \quad (2.14)$$

$$\gamma(P', D) < \gamma(P, D) \quad \forall P' \subset P \quad (2.15)$$

Warunek (2.14) nazywany jest *warunkiem spójności*, natomiast (2.15) — *warunkiem minimalności*. Redukt, zdefiniowany w ten sposób, jest więc minimalnym podzbiorem atrybutów warunkowych, dla których nie spada jakość

klasyfikacji. Nie jest to najogólniejsza definicja reduktu (porównaj np. z [33]), ale wystarczająca do dalszych rozważań.

Reduktem lokalnym [31] dla obiektu x nazywamy minimalny zbiór atrybutów $r(x) \subseteq C$ spełniający warunek:

$$xI_{r(x)}y \Rightarrow xI_Dy \quad \forall y \in U \quad (2.16)$$

Oznacza to, że obiekty nierozróżnialne z x za pomocą zbioru atrybutów $r(x)$ nie mogą należeć do różnych klas; innymi słowy, zbiór $r(x)$ jest wystarczający (a zarazem minimalny) do odróżnienia obiektu x od wszystkich obiektów z innych klas.

2.2.4 Reguły decyzyjne

Jeśli zastosujemy pewien algorytm indukcji reguł do dolnych przybliżeń klas decyzyjnych, mówimy, że otrzymane reguły są *pewne*, w odróżnieniu od indukcji z dolnych przybliżeń — w takim przypadku o regułach mówimy, że są *możliwe*. Rozpatruje się również reguły wygenerowane dla brzegów klas, czyli dla przykładów nie mających jednoznacznie określonej wartości atrybutu decyzyjnego (tzw. reguły *przybliżone*), co z kolei prowadzi do przedstawienia części decyzyjnej takich reguł postaci alternatywy (lub podzbioru) kilku możliwych wartości atrybutu decyzyjnego.

Dla wszystkich atrybutów $a_i \in C$ oraz dla każdego obiektu $x \in U$, powyżej opisane reguły przyjmują następującą postać:

- reguły pewne:

$$\text{jeżeli } f(a_1, x) = r_{a_1} \wedge f(a_2, x) = r_{a_2} \wedge \dots \wedge f(a_p, x) = r_{a_p} \text{ to } x \in Cl_t \quad (2.17)$$

- reguły możliwe:

$$\begin{aligned} \text{jeżeli } f(a_1, x) = r_{a_1} \wedge f(a_2, x) = r_{a_2} \wedge \dots \wedge f(a_p, x) = r_{a_p} \\ \text{to } x \text{ może należeć do } Cl_t \end{aligned} \quad (2.18)$$

- reguły przybliżone:

$$\begin{aligned} \text{jeżeli } f(a_1, x) = r_{a_1} \wedge f(a_2, x) = r_{a_2} \wedge \dots \wedge f(a_p, x) = r_{a_p} \\ \text{to } x \subseteq \{Cl_1, \dots, Cl_n\} \end{aligned} \quad (2.19)$$

Przy generowaniu reguł, wyznaczaniu reduktów i innych algorytmach dotyczących tablicy informacyjnej znajduje zastosowanie *macierz rozróżnialności*, którą definiujemy dla zbioru $P \subseteq C$ jako:

$$\mathcal{M}(P) = \{\delta(x, y) : x, y \in U\} \quad (2.20)$$

gdzie

$$\delta(x, y) = \{a \in P : \neg x I_a y\} \quad (2.21)$$

czyli na pozycji (x, y) znajduje się podzbiór atrybutów ze zbioru P , na których obiekt x i y mają różne wartości. Zauważmy, że $\mathcal{M}(P)$ jest macierzą symetryczną oraz na przekątnej ma zbiory puste.

2.3 Podejście do zbiorów przybliżonych oparte na relacji dominacji

Jeśli dziedzinę danego atrybutu można uporządkować pod względem preferencji, atrybut taki nazywać będziemy *kryterium*. Jest to dodatkowa informacja dotycząca obiektów z tablicy decyzyjnej, ale również dodatkowe potencjalne źródło niespójności. Klasyczna teoria zbiorów przybliżonych nie bierze pod uwagę skali preferencji, ponieważ oparta jest na relacji nierozróżnialności [16]. Może pojawić się sytuacja, w której obiekt x o wartościach na wszystkich kryteriach wyższych od wartości obiektu y zostanie mimo to zakwalifikowany do klasy niższej (gorszej), gdyż był od obiektu y rozróżnialny. Stąd pojawił się pomysł zastąpienia relacji nierozróżnialności relacją dominacji; nowa metodologia została nazwana *podejściem do zbiorów przybliżonych opartym na relacji dominacji* (ang. *Dominance-Based Rough Set Approach, DRSA*) [16, 18, 17].

2.3.1 Relacja dominacji, dolne i górne przybliżenie

Rozpatrzmy przypadek, w którym zbiór atrybutów warunkowych C składa z kryteriów, przy czym można zdefiniować względem każdego kryterium $q \in C$ relację przewyższania (ang. *outranking relation*) S_q na zbiorze U , mającą własności porządku zupełnego, przy czym $x S_q y$ oznacza, że obiekt x jest co najmniej tak dobry jak y względem kryterium q . Relację dominacji D_P (ang. *dominance relation*) na zbiorze $P \subseteq C$ definiujemy w następujący sposób:

$$x D_P y \Leftrightarrow \forall q \in P \ x S_q y \quad (2.22)$$

Relacja dominacji jest zwrotna i przechodnia. Zdefiniujemy zbiór P -dominujący jako $D_P^+(x) = \{y \in U : y D_P x\}$ oraz zbiór P -zdominowany jako $D_P^-(x) = \{y \in U : x D_P y\}$. Zbiory te zastąpią klasyczne granule (zbiory P -elementarne).

Rozważmy zbiór $X \subseteq U$ oraz dwa zbiory $X^{\geq} \subseteq U$ oraz $X^{\leq} \subseteq U$ zawierające, prócz zbioru X dodatkowo obiekty odpowiednio bardziej lub mniej preferowane od obiektów rozważanego zbioru. Możemy teraz przedefiniować B -dolne i B -górne przybliżenie, tym razem definiowane dla zbiorów X^{\geq} i X^{\leq} :

$$\underline{B}(X^{\geq}) = \{x \in U : D_B^+(x) \subseteq X^{\geq}\} \quad (2.23)$$

$$\overline{B}(X^{\geq}) = \bigcup_{x \in X^{\geq}} D_B^+(x) \quad (2.24)$$

$$\underline{B}(X^{\leq}) = \{x \in U : D_B^-(x) \subseteq X^{\leq}\} \quad (2.25)$$

$$\overline{B}(X^{\leq}) = \bigcup_{x \in X^{\leq}} D_B^-(x) \quad (2.26)$$

Zwróćmy uwagę, że rozpatrujemy dwa górne i dwa dolne przybliżenia związane z każdym zbiorem (dla dwóch kumulacji klas) z powodu utraty symetrii przez relację dominacji. B -brzegiem, oznaczanym przez $BN_B(X)$, nazywamy różnicę między B -górnym i B -dolnym przybliżeniem.

Założmy ponadto, że rodzina zbiorów $\mathbf{Cl} = \{Cl_1, Cl_2, \dots, Cl_n\}$ jest partycją zbioru U na uporządkowane klasy decyzyjne, przy czym każdy obiekt należy do dokładnie jednej klasy, a obiekty z każdej Cl_s są preferowane nad obiekty z klasy Cl_t o ile $s > t$. Dodatkowo zdefiniujemy zbiory odpowiadające skumulowanym klasom decyzyjnym: $Cl_t^{\geq} = \bigcup_{s \geq t} Cl_s$ oraz $Cl_t^{\leq} = \bigcup_{s \leq t} Cl_s$. Wtedy przy pomocy zbioru kryteriów C możemy utworzyć górne i dolne przybliżenia dla takich kumulacji: $\underline{C}(Cl_t^{\geq})$, $\overline{C}(Cl_t^{\geq})$, $\underline{C}(Cl_t^{\leq})$ oraz $\overline{C}(Cl_t^{\leq})$.

2.3.2 Jakość klasyfikacji, redukt

Przeddefiniować należy również pojęcia jakości klasyfikacji, mianowicie:

$$\gamma(P, Cl) = \frac{|U - \bigcup_{Cl_t \in Cl} BN_P(Cl_t^{\geq})|}{|U|} \quad (2.27)$$

Współczynnik ten mówi, jaka część spośród wszystkich obiektów jest poprawnie (jednoznacznie) zaklasyfikowana.

Pojęcie reduktu jest analogiczne do przypadku klasycznego podejścia do zbiorów rozmytych (podrozdział 2.2.3), natomiast redukt lokalny rozpada się na dwa rodzaje: lokalny dominujący ($r^+(x)$, dla $x \in Cl_t^{\geq}$) i lokalny zdominowany ($r^-(x)$, dla $x \in Cl_t^{\leq}$) [5]. Oba muszą być minimalne w sensie zawierania, a zdefiniowane są dla w następujący sposób:

$$\forall y \in Cl_{t-1}^{\leq} \neg y D_{r^+(x)} x \quad (2.28)$$

$$\forall y \in Cl_{t+1}^{\geq} \neg x D_{r^-(x)} y \quad (2.29)$$

2.3.3 Reguły decyzyjne

Podobnie, jak w przypadku klasycznym, wyróżniamy reguły pewne, możliwe i przybliżone. Ze względu na brak symetrii relacji dominacji wyróżnia się jednak rodzaje reguł. D_{\geq} -reguła, wyindukowana z kumulacji klasy w górę Cl_t^{\geq} ma postać:

$$\text{jeżeli } (f(q_1, x) \geq r_{q_1}) \wedge \dots \wedge (f(q_r, x) \geq r_{q_r}) \text{ to } x \in Cl_t^{\geq} \quad (2.30)$$

Z kolei D_{\leq} -reguła, wyindukowana z kumulacji klasy w dół Cl_t^{\leq} ma postać:

$$\text{jeżeli } (f(q_1, x) \leq r_{q_1}) \wedge \dots \wedge (f(q_r, x) \leq r_{q_r}) \text{ to } x \in Cl_t^{\leq} \quad (2.31)$$

Natomiast przybliżona $D_{\geq\leq}$ -reguła, wyindukowana z $\overline{B}(Cl^{\leq s}) \cap \underline{B}(Cl^{\geq t})$, $s < t$, ma postać:

$$\begin{aligned} &\text{jeżeli } (f(q_1, x) \geq r_{q_1}) \wedge \dots \wedge (f(q_k, x) \geq r_{q_k}) \wedge (f(q_{k+1}, x) \leq r_{q_{k+1}}) \wedge \\ &\wedge \dots \wedge (f(q_r, x) \leq r_{q_r}) \text{ to } x \in Cl_s \cup \dots \cup Cl_t \end{aligned} \quad (2.32)$$

Macierz dominacji, którą definiujemy dla zbioru $P \subseteq C$, ma postać:

$$\mathcal{D}(P) = \{\delta^{\geq}(x, y) : x, y \in U\} \quad (2.33)$$

gdzie:

$$\delta^{\geq}(x, y) = \{q \in P : xS_q y\} \quad (2.34)$$

Zwróćmy uwagę, że macierz $\mathcal{D}(P)$ nie jest symetryczna.

Rozdział 3

Reguły gradualne

3.1 Wprowadzenie

Termin *reguły gradualne* pojawia się już w artykułach Dubois i Prade [7, 9] i dotyczy typu reguł rozmytych, odpowiadającym stwierdzeniu: „im bardziej x jest A , tym bardziej y jest B ” lub „im bardziej x jest B , tym mniej y jest B ”, gdzie x i y są obiektami, a A oraz B — zbiorami rozmytymi. Taki sposób sformułowania reguł narzuca warunek monotoniczności na zależność między przesłanką a konkluzją reguły tak, aby wzrost przynależności x do A odpowiadał zawsze równoczesnemu wzrostowi (spadkowi) przynależności y do zbioru B . Jeśli przez $\mu_A(x)$ i $\mu_B(x)$ oznaczymy miary przynależności obiektu x do wymienionych zbiorów, zwykle regułę gradualną można zapisać w postaci: „jeśli $\mu_A(x) \geq \alpha$, to $\mu_B(x) \geq \alpha$ ” lub „jeśli $\mu_A(x) \geq \alpha$, to $\mu_B(x) \leq \alpha$ ”

S. Greco, M. Inuiguchi i R. Słowiński przedstawili nowe podejście do opisu danych za pomocą reguł gradualnych, bazujące, oprócz logiki rozmytej, również na teorii zbiorów przybliżonych [14, 15, 13]. Reguły tam rozpatrywane są ogólniejsze od typowych reguł gradualnych, mają bowiem postać: „jeśli $\mu_A(x) \geq \alpha$, to $\mu_B(x) \geq \beta$ ” (pozytywny związek) lub „jeśli $\mu_A(x) \geq \alpha$, to $\mu_B(x) \leq \beta$ ” (negatywny związek). Poprzez zastosowanie reguł o takiej postaci zbiór danych może zostać opisany opierając się przede wszystkim o pojęcie monotoniczności. Tym samym podstawową rolę pełni tu relacja porządku elementów na skali preferencji kryteriów. Co więcej, podstawowym założeniem metodyki jest pełna rezygnacja z używania wartości liczbowych, jakie przyjmują obiekty na kryteriach, czyli rezygnacja z porównywania odległości między obiektami. Motywacją do takiego podejścia jest fakt, że stwierdzenia postaci: „obiekt x należy do zbioru X dwukrotnie bardziej prawdopodobnie niż obiekt y ” dla człowieka są mało znaczące, a w niektórych przypadkach nie mają wręcz sensu. W ten sposób funkcje przynależności, mimo numerycznego charakteru, przechowują jedynie informacje porządkowe i tylko na nich można oprzeć późniejsze ich przetwarzanie i wnioskowanie, co prowadzi do rezygnacji z typowych t-norm i s-norm stosowanych w przetwarzaniu rozmytym. Rzeczywiście, porządek wyników po

zastosowaniu klasycznych t- i s-norm zmienia się przy zmianie wyłącznie wartości argumentów, a zachowaniu porządku między nimi[14].

3.1.1 Reguły decyzyjne z monotoniczną zależnością między przesłanką a konkluzją

Reguły gradualne cechuje monotoniczna zależność między częścią warunkową a decyzyjną. Tym samym rozważane reguły będą miały postać, podkreślającą tę cechę:

- *reguła dolnego przybliżenia z pozytywnym związkiem* (LP-reguła): „jeśli warunek X ma wiarygodność $C(X) \geq \alpha$, to decyzja Y ma wiarygodność $C(Y) \geq \beta$ ”;
- *reguła dolnego przybliżenia z negatywnym związkiem* (LN-reguła): „jeśli warunek X ma wiarygodność $C(X) \leq \alpha$, to decyzja Y ma wiarygodność $C(Y) \geq \beta$ ”;
- *reguła górnego przybliżenia z pozytywnym związkiem* (UP-reguła): „jeśli warunek X ma wiarygodność $C(X) \leq \alpha$, to decyzja Y może mieć wiarygodność $C(Y) \leq \beta$ ”;
- *reguła górnego przybliżenia z negatywnym związkiem* (UN-reguła): „jeśli warunek X ma wiarygodność $C(X) \geq \alpha$, to decyzja Y może mieć wiarygodność $C(Y) \leq \beta$ ”.

Powyższe sformułowania można reprezentować za pomocą uporządkowanych trójek, odpowiednio: $\langle X, Y, f^+ \rangle$, $\langle X, Y, f^- \rangle$, $\langle X, Y, g^+ \rangle$ oraz $\langle X, Y, g^- \rangle$, gdzie funkcje f^+ , f^- , g^+ , $g^- : [0, 1] \rightarrow [0, 1]$ wyrażają zależność między wiarygodnością X , a wiarygodnością Y . Stąd, używając powyższych funkcji, można zapisać reguły w sposób nieco bardziej formalny:

- LP-reguła: „jeśli warunek X ma wiarygodność $C(X) \geq \alpha$, to decyzja ma wiarygodność $C(Y) \geq \beta = f^+(\alpha)$ ”;
- LN-reguła: „jeśli warunek X ma wiarygodność $C(X) \leq \alpha$, to decyzja ma wiarygodność $C(Y) \geq \beta = f^-(\alpha)$ ”;
- UP-reguła: „jeśli warunek X ma wiarygodność $C(X) \leq \alpha$, to decyzja ma wiarygodność $C(Y) \leq \beta = g^+(\alpha)$ ”;
- UN-reguła: „jeśli warunek X ma wiarygodność $C(X) \geq \alpha$, to decyzja ma wiarygodność $C(Y) \leq \beta = g^-(\alpha)$ ”.

Zwróćmy uwagę, że składnia powyższych reguł jest podobna do składni reguł $D \geq$ oraz $D \leq$ (opisanych w części 2.3.3), indukowanych z dolnych i górnych przybliżeń kumulacji klas [16]. Analogie między oboma podejściami są silne i będą widoczne w dalszym opisie.

3.1.2 Rozszerzenie na reguły z wieloma przesłankami

Powyższe reguły mają część warunkową, która składa się z pojedynczej przesłanki, co utrudnia ich praktyczne zastosowania. Wprowadzono [15] naturalne uogólnienie na reguły z wieloma warunkami (przesłankami):

- *reguły dolnego przybliżenia* (L-reguły):
 „jeśli $x \in X_{i_1}^\uparrow$ z wiarygodnością $C(X_{i_1}^\uparrow) \geq \alpha_{i_1} \wedge x \in X_{i_2}^\uparrow$ z wiarygodnością $C(X_{i_2}^\uparrow) \geq \alpha_{i_2} \wedge \dots \wedge x \in X_{i_r}^\uparrow$ z wiarygodnością $C(X_{i_r}^\uparrow) \geq \alpha_{i_r} \wedge$
 $x \in X_{j_1}^\downarrow$ z wiarygodnością $C(X_{j_1}^\downarrow) \leq \alpha_{i_1} \wedge x \in X_{j_2}^\downarrow$ z wiarygodnością $C(X_{j_2}^\downarrow) \leq \alpha_{i_2} \wedge \dots \wedge x \in X_{j_s}^\downarrow$ z wiarygodnością $C(X_{j_s}^\downarrow) \leq \alpha_{i_s}$
 to $x \in Y$ ma wiarygodność $C(Y) \geq \beta = f(\alpha_{i_1}, \dots, \alpha_{i_r}, \alpha_{j_1}, \dots, \alpha_{j_s})$.
- *reguły górnego przybliżenia* (U-reguły):
 „jeśli $x \in X_{i_1}^\uparrow$ z wiarygodnością $C(X_{i_1}^\uparrow) \leq \alpha_{i_1} \wedge x \in X_{i_2}^\uparrow$ z wiarygodnością $C(X_{i_2}^\uparrow) \leq \alpha_{i_2} \wedge \dots \wedge x \in X_{i_r}^\uparrow$ z wiarygodnością $C(X_{i_r}^\uparrow) \leq \alpha_{i_r} \wedge$
 $x \in X_{j_1}^\downarrow$ z wiarygodnością $C(X_{j_1}^\downarrow) \geq \alpha_{i_1} \wedge x \in X_{j_2}^\downarrow$ z wiarygodnością $C(X_{j_2}^\downarrow) \geq \alpha_{i_2} \wedge \dots \wedge x \in X_{j_s}^\downarrow$ z wiarygodnością $C(X_{j_s}^\downarrow) \geq \alpha_{i_s}$
 to $x \in Y$ ma wiarygodność $C(Y) \leq \beta = g(\alpha_{i_1}, \dots, \alpha_{i_r}, \alpha_{j_1}, \dots, \alpha_{j_s})$.

Przez $X_{i_1}^\uparrow$ oznaczamy rozmyte wartości atrybutów będące w pozytywnym związku z decyzją Y , natomiast przez $X_{i_1}^\downarrow$ — rozmyte wartości atrybutów w związku negatywnym. Ponieważ oba typy zbiorów rozmytych występują w części warunkowej, rozróżnienie funkcji $f(\cdot)$ na $f^+(\cdot)$ i $f^-(\cdot)$ oraz funkcji $g(\cdot)$ na $g^+(\cdot)$ i $g^-(\cdot)$, staje się zbędne.

3.2 Podejście rozmyto-przybliżone do indukcji reguł

W przedstawionych powyżej regułach pojawia się pojęcie wiarygodności spełniania warunku X przez obiekt x , równoznaczne z wartością funkcji przynależności obiektu x do zbioru rozmytego X . Poprzez zdefiniowanie odpowiednich funkcji przynależności jesteśmy w stanie przekształcić oryginalną tablicę decyzyjną w tablicę zawierającą stopnie przynależności obiektów na atrybutach. Przyjmując opis zbioru za pomocą reguł z zależnością monotoniczną, musimy równocześnie przyjąć pewien kierunek preferencji dla każdego zbioru rozmytego opartego na każdym z atrybutów, co oznacza, że stopnie przynależności do zbiorów opartych na atrybutach mają charakter kryteriów. Tym samym mamy do czynienia z niespójnościami podobnego typu, jak w przypadku DRSA (punkt 2.3). Należy zapewnić, aby obiekt x lepszy globalnie od obiektu y (tzn. o stopniach przynależności do zbiorów X^\uparrow większych, niż y , a o stopniach przynależności do X^\downarrow mniejszych, niż y) miał nie mniejszą wartość wiarygodności decyzji Y w L-regułach oraz nie większą w U-regułach.

3.2.1 Przypadek dla reguł z jedną przesłanką

Przypadek reguł z jedną przesłanką, czyli przypadek z jednym kryterium jest dużo przejrzystszy od przypadku wielokryterialnego. Przyjmijmy, że $\mu_X: U \rightarrow [0, 1]$ i $\mu_Y: U \rightarrow [0, 1]$ są funkcjami przynależności odpowiednio do zbioru X opartego na atrybucie warunkowym i zbioru Y opartego o decyzję. Dla związku „im bardziej x jest X , tym bardziej y jest Y ” definiujemy dolne przybliżenie Y oparte na informacji o X jako zbiór rozmyty $\underline{App}^+(X, Y)$, a funkcję przynależności do tego zbioru dla każdego obiektu $x \in U$ jako:

$$\mu[\underline{App}^+(X, Y), x] = \inf_{z \in U: \mu_X(z) \geq \mu_X(x)} \{\mu_Y(z)\} \quad (3.1)$$

W przypadku, gdy istnieją obiekty $x, z \in U$ takie, że $\mu_X(x) < \mu_X(z)$ i równocześnie $\mu_Y \geq \rho$ podczas gdy $\mu_Y < \rho$, mamy do czynienia z niespójnością. Jednak, wykorzystując wiedzę o X oraz pozytywną zależność między X i Y , możemy powiedzieć, że obiekt x należy do zbioru rozmytego Y , nie powodując niespójności, z wiarygodnością $\rho \leq \mu[\underline{App}^+(X, Y), x]$.

Podobnie, górne przybliżenie Y oparte na informacji o X definiujemy jako zbiór rozmyty $\overline{App}^+(X, Y)$, a funkcję przynależności do tego zbioru dla każdego $x \in U$ jako:

$$\mu[\overline{App}^+(X, Y), x] = \sup_{z \in U: \mu_X(z) \leq \mu_X(x)} \{\mu_Y(z)\} \quad (3.2)$$

Ponownie, w przypadku niespójności dla obiektów x i z , wykorzystując wiedzę o X oraz pozytywną zależność między X i Y , obiekt z nie powodowałby niespójności należąc do zbioru rozmytego Y z wiarygodnością $\omega \geq \mu[\overline{App}^+(X, Y), z]$.

Podobnie możemy zdefiniować dolne i górne przybliżenie Y oparte na informacji o X dla związku „im bardziej x jest X , tym mniej y jest Y ”, jako zbiory rozmyte $\underline{App}^-(X, Y)$ i $\overline{App}^-(X, Y)$ o funkcjach przynależności:

$$\mu[\underline{App}^-(X, Y), x] = \inf_{z \in U: \mu_X(z) \leq \mu_X(x)} \{\mu_Y(z)\} \quad (3.3)$$

$$\mu[\overline{App}^-(X, Y), x] = \sup_{z \in U: \mu_X(z) \geq \mu_X(x)} \{\mu_Y(z)\} \quad (3.4)$$

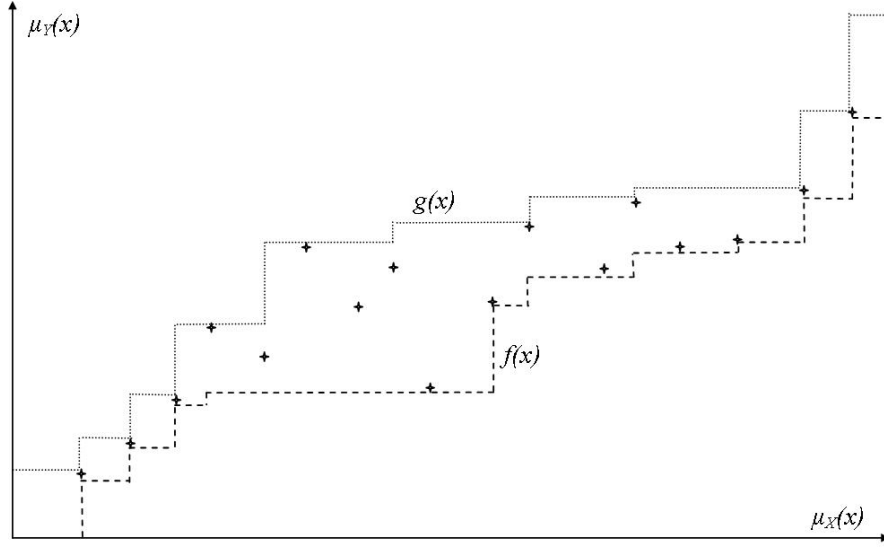
Między przybliżeniami zachodzą relacje:

$$\mu[\underline{App}^+(X, Y), x] \leq \mu_Y(x) \leq \mu[\overline{App}^+(X, Y), x] \quad (3.5)$$

$$\mu[\underline{App}^-(X, Y), x] \leq \mu_Y(x) \leq \mu[\overline{App}^-(X, Y), x] \quad (3.6)$$

odpowiadające związkom zachodzącym w klasycznej teorii zbiorów przybliżonych [26].

Górne i dolne przybliżenia pozwalają na znajdowanie pewnych i przybliżonych reguł decyzyjnych o postaci opisanej w punkcie 3.1.1. Utworzenie takich



Rysunek 3.1: Przykładowy zbiór danych oraz funkcje dolnego i górnego przybliżenia — $f^+(\cdot)$ (linia przerywana) i $g^+(\cdot)$ (linia kropkowana)

reguł jest równoznaczne z utworzeniem funkcji $f^+(\cdot)$, $f^-(\cdot)$, $g^+(\cdot)$ oraz $g^-(\cdot)$, które definiujemy w następujący sposób:

$$f^+(\alpha) = \sup_{z \in U: \mu_X(x) \leq \alpha} \{\mu[\underline{App}^+(X, Y), x]\} \quad (3.7)$$

$$f^-(\alpha) = \sup_{z \in U: \mu_X(x) \geq \alpha} \{\mu[\underline{App}^-(X, Y), x]\} \quad (3.8)$$

$$g^+(\alpha) = \inf_{z \in U: \mu_X(x) \geq \alpha} \{\mu[\overline{App}^+(X, Y), x]\} \quad (3.9)$$

$$g^-(\alpha) = \inf_{z \in U: \mu_X(x) \leq \alpha} \{\mu[\overline{App}^-(X, Y), x]\} \quad (3.10)$$

Przykładowy zbiór danych, dla którego zachodzi pozytywna zależność między wiarygodnością X , a wiarygodnością Y , oraz funkcję $f^+(\cdot)$ i $g^+(\cdot)$ zostały przedstawione na rys. 3.1.

Ponownie pojawia się analogia do DRSA. Zauważmy, że funkcje $f^+(\cdot)$, $f^-(\cdot)$, $g^+(\cdot)$ oraz $g^-(\cdot)$ spełniają podobną rolę, co (odpowiednio) dolne przybliżenie kumulacji klas w górę, dolne przybliżenie kumulacji klas w dół, oraz górne przybliżenie kumulacji klas w górę i w dół.

3.2.2 Przypadek dla reguł z wieloma przesłankami

Powyższe rozważania da się oczywiście uogólnić na reguły o wielu warunkach, czyli problem wielokryterialny. W takim przypadku dolne przybliżenie Y oparte na informacji o $X_{i_1}^\uparrow, \dots, X_{i_r}^\uparrow, X_{j_1}^\downarrow, \dots, X_{j_s}^\downarrow$ definiujemy jako zbiór rozmyty $\underline{App}(X_{i_1}^\uparrow, \dots, X_{i_r}^\uparrow, X_{j_1}^\downarrow, \dots, X_{j_s}^\downarrow, Y)$, którego funkcja przynależności dla każdego obiektu $x \in U$ równa jest:

$$\mu[\underline{App}(X_{i_1}^\uparrow, \dots, X_{i_r}^\uparrow, X_{j_1}^\downarrow, \dots, X_{j_s}^\downarrow, Y), x] = \inf_{z \in D^\uparrow(x)} [\mu_Y(z)] \quad (3.11)$$

gdzie dla każdego $x \in U$, $D^\uparrow(x)$ jest niepustym zbiorem definiowanym następująco:

$$D^\uparrow(x) = \{z \in U : \mu_{X_h}(z) \geq \mu_{X_h}(x) \forall X_h \in \{X_{i_1}^\uparrow, \dots, X_{i_r}^\uparrow\} \wedge \wedge \mu_{X_h}(z) \leq \mu_{X_h}(x) \forall X_h \in \{X_{j_1}^\downarrow, \dots, X_{j_s}^\downarrow\}\}. \quad (3.12)$$

Zbiór $D^\uparrow(x)$ jest odpowiednikiem zbioru $D^+(x)$ (punkt 2.3) w przypadku podejścia do zbiorów rozmytych opartego na dominacji (DRSA). Tym samym $D^\uparrow(x)$ może być interpretowany jako zbiór wszystkich obiektów dominujących obiekt x . Stąd powyższa definicja przynależności do dolnego przybliżenia dotyczy niespójności zachodzących w przypadku, gdy obiekt z dominuje obiekt x , a zarazem wiarygodność przynależności z do Y (decyzji) jest mniejsza, niż wiarygodność przynależności x do Y .

Podobnie, górne przybliżenie Y oparte na informacji o $X_{i_1}^\uparrow, \dots, X_{i_r}^\uparrow, X_{j_1}^\downarrow, \dots, X_{j_s}^\downarrow$ definiujemy jako zbiór rozmyty $\overline{App}(X_{i_1}^\uparrow, \dots, X_{i_r}^\uparrow, X_{j_1}^\downarrow, \dots, X_{j_s}^\downarrow, Y)$, którego funkcja przynależności dla każdego obiektu $x \in U$ równa jest:

$$\mu[\overline{App}(X_{i_1}^\uparrow, \dots, X_{i_r}^\uparrow, X_{j_1}^\downarrow, \dots, X_{j_s}^\downarrow, Y), x] = \sup_{z \in D^\downarrow(x)} [\mu_Y(z)] \quad (3.13)$$

gdzie dla każdego $x \in U$, $D^\downarrow(x)$ jest niepustym zbiorem definiowanym następująco:

$$D^\downarrow(x) = \{z \in U : \mu_{X_h}(z) \leq \mu_{X_h}(x) \forall X_h \in \{X_{i_1}^\uparrow, \dots, X_{i_r}^\uparrow\} \wedge \wedge \mu_{X_h}(z) \geq \mu_{X_h}(x) \forall X_h \in \{X_{j_1}^\downarrow, \dots, X_{j_s}^\downarrow\}\}. \quad (3.14)$$

Zbiór $D^\downarrow(x)$ jest odpowiednikiem zbioru $D^-(x)$ znanego z metodyki DRSA, stąd $D^\downarrow(x)$ może być interpretowany jako zbiór wszystkich obiektów zdominowanych przez obiekt x .

Podobnie, jak w przypadku jednokryterialnym, zachodzi związek:

$$\begin{aligned} \mu[\underline{App}(X_{i_1}^\uparrow, \dots, X_{i_r}^\uparrow, X_{j_1}^\downarrow, \dots, X_{j_s}^\downarrow, Y), x] &\leq \mu_Y(x) \leq \\ &\leq \mu[\overline{App}(X_{i_1}^\uparrow, \dots, X_{i_r}^\uparrow, X_{j_1}^\downarrow, \dots, X_{j_s}^\downarrow, Y), x] \end{aligned} \quad (3.15)$$

Ponieważ w definicji dolnego i górnego przybliżenia występują zarówno zbiory, dla których wiarygodność znajduje się w pozytywnym związku z decyzją Y ($X_{i_1}^\uparrow$), jak i zbiory z wiarygodnością w negatywnym związku z Y ($X_{i_1}^\downarrow$), więc nie rozróżnia się dwóch górnych i dolnych przybliżeń, $\underline{App}^+(X, Y)$, $\underline{App}^-(X, Y)$, $\overline{App}^+(X, Y)$ i $\overline{App}^-(X, Y)$, tak jak to było w przypadku jednokryterialnym.

Reguły (funkcje) przyjmują w tym przypadku następującą postać:

$$f(\alpha) = \sup_{x \in D^\downarrow(\alpha)} \{\mu[\underline{App}(X_{i_1}^\uparrow, \dots, X_{i_r}^\uparrow, X_{j_1}^\downarrow, \dots, X_{j_s}^\downarrow, Y), x]\} \quad (3.16)$$

$$g(\alpha) = \inf_{x \in D^\uparrow(\alpha)} \{\mu[\overline{App}(X_{i_1}^\uparrow, \dots, X_{i_r}^\uparrow, X_{j_1}^\downarrow, \dots, X_{j_s}^\downarrow, Y), x]\} \quad (3.17)$$

gdzie $\alpha = [\alpha_{i_1}, \dots, \alpha_{i_r}, \alpha_{j_1}, \dots, \alpha_{j_s}]$ reprezentuje punkt w przestrzeni kryteriów.

3.2.3 Zasada maksymalnej ostrożności

Definicje (3.16) i (3.17) nie są jedynymi możliwymi definicjami funkcji $f(\cdot)$ i $g(\cdot)$ — można je zdefiniować dość dowolnie. Wymaga się jedynie, by spełniały pewne założenia dotyczące spójności, mianowicie:

- dla każdego $\alpha \in [0, 1]^{r+s}$, dla wszystkich obiektów $x \in U$, takich że $x \in D^\downarrow(\alpha)$ musi zachodzić:

$$f(\alpha) \geq \mu[\underline{App}(X_{i_1}^\uparrow, \dots, X_{i_r}^\uparrow, X_{j_1}^\downarrow, \dots, X_{j_s}^\downarrow, Y), x] \quad (3.18)$$

$$g(\alpha) \leq \mu[\overline{App}(X_{i_1}^\uparrow, \dots, X_{i_r}^\uparrow, X_{j_1}^\downarrow, \dots, X_{j_s}^\downarrow, Y), x] \quad (3.19)$$

- dla każdego $\alpha \in [0, 1]^{r+s}$, dla wszystkich obiektów $x \in U$, takich że $x \in D^\uparrow(\alpha)$ musi zachodzić:

$$f(\alpha) \leq \mu[\underline{App}(X_{i_1}^\uparrow, \dots, X_{i_r}^\uparrow, X_{j_1}^\downarrow, \dots, X_{j_s}^\downarrow, Y), x] \quad (3.20)$$

$$g(\alpha) \geq \mu[\overline{App}(X_{i_1}^\uparrow, \dots, X_{i_r}^\uparrow, X_{j_1}^\downarrow, \dots, X_{j_s}^\downarrow, Y), x] \quad (3.21)$$

Każdą regułę (funkcję) spełniającą powyższe warunki, nazwiemy *poprawną*. Dodatkowo, aby znaleźć konkretną postać reguły, definiuje się [15] pojęcie *ostrożności*: w przypadku dwóch L-reguł, $f(\cdot)$ jest bardziej ostrożna, niż $f'(\cdot)$, gdy dla każdego $\alpha \in [0, 1]^{r+s}$ zachodzi $f(\alpha) \leq f'(\alpha)$. Podobnie, dla dwóch U-reguł, $g(\cdot)$ jest bardziej ostrożna, niż $g'(\cdot)$, gdy dla każdego $\alpha \in [0, 1]^{r+s}$ zachodzi $g(\alpha) \geq g'(\alpha)$. Powyższy warunek mówi, że ostrożniejsza reguła to taka, która wyciąga bardziej konserwatywne wnioski z danych. Innymi słowy, najostrożniejsza reguła szacuje wartości w taki sposób, aby wyciągać jedynie w pełni uzasadnione przez dane wnioski i nic ponadto.

Można pokazać, że reguły zdefiniowane w (3.16) i (3.17), są regułami możliwie najbardziej ostrożnymi.

3.2.4 Wyznaczanie wiarygodności przynależności obiektów do dolnych i górnych przybliżeń

Można przyjąć, że dla rzeczywistych problemów i zbiorów danych, zbiór obiektów U ma skończoną liczbę. Tym samym, jeśli liczba obiektów jest skończona, w równaniach (3.1), (3.2), (3.3), (3.4), (3.11) i (3.13) można zastąpić szukanie kresów dolnych i górnych (sup i inf) szukaniem największych i najmniejszych elementów w zbiorze (max i min), co czyni problem łatwiejszym do implementacji. W tym wypadku, aby wyznaczyć wiarygodność dolnego przybliżenia dla danego obiektu x , rozważamy wszystkie obiekty dominujące x (łącznie z samym x), czyli należące do zbioru $D^\uparrow(x)$. Szukaną wartością będzie największa wartość wiarygodności decyzji dla przeglądanych obiektów. W szczególności, jeśli zbiór jest spójny, największą spośród tych wartości będzie wiarygodność decyzji dla samego x . Podobnie, przy wyznaczaniu wiarygodności dolnego przybliżenia dla obiektu x , przeglądamy wszystkie obiekty zdominowane przez x , czyli należące do zbioru $D^\downarrow(x)$, a szukaną wartością będzie najmniejsza z wiarygodności decyzji dla wszystkich obiektów z tego zbioru. Ponownie, w spójnym zbiorze, wartością tą będzie wiarygodność decyzji dla samego x .

Wyznaczanie dolnych i górnych przybliżeń ułatwia posortowanie obiektów względem wiarygodności decyzji. W takim przypadku, przy wyznaczaniu dolnego przybliżenia dla x wystarczy przeglądać tylko obiekty o większej wiarygodności decyzji (leżące na liście za obiektem x), a po napotkaniu pierwszego, który dominuje x , można mieć pewność, że szukaną wartością jest mniejsza z wiarygodności decyzji: obiektu x lub obiektu znalezionego. Podobnie, dla górnego przybliżenia, ograniczamy się do przeglądania obiektów o mniejszej wiarygodności i kończymy szukanie na pierwszym obiekcie zdominowanym przez x . Powyższe działania powodują zwiększenie szybkości obliczeń, wymagają jednak zarazem posortowania zbioru obiektów.

3.3 Wnioskowanie rozmyto-przybliżone

Utworzone reguły z jednej strony mają wyjaśnić zbiór danych, a z drugiej strony mają służyć do klasyfikacji nowych obiektów, która odbywa się poprzez wnioskowanie. Wyróżniamy dwie podstawowe reguły wnioskowania — *modus ponens* (MP), zwane wnioskowaniem wprzód, oraz *modus tolens* (MT), zwane wnioskowaniem w tył.

Klasyczny MP ma następującą postać:

$$\begin{array}{lcl} \text{jeśli} & X \rightarrow Y & \text{jest prawdą} \\ \text{i} & X & \text{jest prawdą} \\ \hline \text{to} & Y & \text{jest prawdą} \end{array} \quad (3.22)$$

MP ma następującą interpretację: przyjmując, że prawdziwa jest implikacja $X \rightarrow Y$ oraz fakt X (przesłanka), otrzymujemy kolejny fakt Y (konkluzję).

Podobnie możemy zdefiniować MT:

$$\begin{array}{lcl}
\text{jeśli} & X \rightarrow Y & \text{jest prawdą} \\
\text{i} & Y & \text{jest fałszywe} \\
\hline
\text{to} & X & \text{jest fałszywe}
\end{array} \quad (3.23)$$

Interpretacja MT jest następująca: przyjmując, że prawdziwa jest implikacji $X \rightarrow Y$ oraz, że fakt Y (konkluzja) jest fałszywa, wnioskujemy, że fakt X (przesłanka) również musi być fałszywa, czyli otrzymujemy kolejny fakt $\neg X$.

Obok klasycznego *modus ponens*, rozważa się również wnioskowanie w oparciu o rozmytą przesłankę. Dalszym uogólnieniem MP jest próba wnioskowania na podstawie implikacji $X \rightarrow Y$, wiedząc, że prawdziwa jest rozmyta przesłanka X' — w wyniku otrzymujemy rozmytą konkluzję Y' . Można również rozpatrywać wnioskowanie w oparciu o rozmytą relację porządku \succeq_X ; w tym przypadku przesłanka ma postać: $\mu_X(x) \succeq_X \alpha$. Rozważanie wszystkich powyższych zagadnień, osobno dla przypadku jednej i wielu przesłanek, wykracza poza zakres niniejszej pracy, gdyż zagadnienia wnioskowania rozmytego nie są podstawowym problemem w niej rozważanym. W związku z tym ograniczymy się do przeanalizowania wnioskowania rozmytego dla przypadku jednokryterialnego w dwóch wariantach: ze ścisłą i rozmytą przesłanką. Informacje na temat pozostałych zagadnień można znaleźć w [13].

3.3.1 Wnioskowanie w oparciu o ścisłą przesłankę

Zastępując klasyczną implikację przez regułę decyzyjną, otrzymujemy cztery rodzaje rozmytego MP:

- wnioskowanie dla LP-reguły (LP-MP):

$$\begin{array}{lcl}
\text{jeśli} & \mu_X(x) \geq \alpha \rightarrow \mu_Y(x) \geq f^+(\alpha) \\
\text{i} & \mu_X(x) = \alpha' \\
\hline
\text{to} & \mu_Y(x) \geq f^+(\alpha')
\end{array} \quad (3.24)$$

- wnioskowanie dla LN-reguły (LN-MP):

$$\begin{array}{lcl}
\text{jeśli} & \mu_X(x) \leq \alpha \rightarrow \mu_Y(x) \geq f^-(\alpha) \\
\text{i} & \mu_X(x) = \alpha' \\
\hline
\text{to} & \mu_Y(x) \geq f^-(\alpha')
\end{array} \quad (3.25)$$

- wnioskowanie dla UP-reguły (UP-MP):

$$\begin{array}{lcl}
\text{jeśli} & \mu_X(x) \leq \alpha \rightarrow \mu_Y(x) \leq g^+(\alpha) \\
\text{i} & \mu_X(x) = \alpha' \\
\hline
\text{to} & \mu_Y(x) \leq g^+(\alpha')
\end{array} \quad (3.26)$$

- wnioskowanie dla UN-reguły (UN-MP):

$$\begin{array}{lcl}
\text{jeśli} & \mu_X(x) \geq \alpha \rightarrow \mu_Y(x) \leq g^+(\alpha) & \\
\text{i} & \mu_X(x) = \alpha' & \\
\hline
\text{to} & \mu_Y(x) \leq g^+(\alpha') & (3.27)
\end{array}$$

Interpretacja powyższych schematów jest następująca: wiedząc, że prawdziwa jest reguła gradualna opisująca związek między X a Y , oraz znając wiarygodność warunku X , możemy wnioskować, jaka co najmniej (co najwyżej) powinna być wiarygodność decyzji Y . Ponieważ funkcja $f(\cdot)$ dla schematów LP-MP oraz LN-MP ogranicza wiarygodność decyzji od dołu, a funkcja $g(\cdot)$ dla schematów UP-MP oraz UN-MP ogranicza wiarygodność decyzji od dołu, jesteśmy w stanie za pomocą obu tych funkcji stwierdzić, że dla dowolnego obiektu x takiego, że $\mu_X(x) = \alpha$, jego wartość wiarygodności Y znajduje się w przedziale $[f(\alpha), g(\alpha)]$. Tym samym możemy za pomocą reguł gradualnych klasyfikować nowe obiekty, przypisując każdemu z nich przedział, w jakim powinna znaleźć się wiarygodność decyzji.

3.3.2 Wnioskowanie w oparciu o rozmytą przesłankę

Przyjmijmy, że przesłanka jest teraz faktem rozmytym, $\mu_X(x) = \tilde{\alpha}$. Innymi słowy, każdej przesłance $\mu_X(x) = \alpha$ przypisana jest miara przynależności do zbioru rozmytego $\tilde{\alpha}$, $\mu_{\tilde{\alpha}}(\alpha)$. Tym samym spodziewamy się, że również konkluzja będzie rozmyta. Powyższy sposób wnioskowania nazywany jest: *rozmyte²-przybliżone modus ponens*.

Skoro konkluzja będzie faktem rozmytym, $\mu_Y(x) = \tilde{\beta}$, należy znaleźć miarę przynależności do $\tilde{\beta}$, $\mu_{\tilde{\beta}}(\beta)$, przypisaną każdej konkluzji $\mu_Y(x) \geq \beta = f^+(\alpha)$. Miara ta z pewnością będzie zależała od $\mu_{\tilde{\alpha}}(\alpha)$, jednak równocześnie musi spełniać następującą zasadę spójności: dla dwóch konkluzji, $\mu_Y(x) \geq \beta_1$ oraz $\mu_Y(x) \geq \beta_2$, jeśli $\beta_1 \leq \beta_2$, to musi zająć $\mu_{\tilde{\beta}}(\beta_1) \geq \mu_{\tilde{\beta}}(\beta_2)$. Innymi słowy, w przypadku schematu LP-MP, funkcja przynależności $\mu_{\tilde{\beta}}(\beta)$ musi być malejąca, gdyż inaczej oznaczałoby to przypisanie silniejszej konkluzji większej wiarygodności.

W ten sposób definiujemy, osobno dla wszystkich czterech schematów wnioskowania MP, funkcję przynależności rozmytej konkluzji $\mu_{\tilde{\beta}}(\cdot)$:

- dla rozmytego²-przybliżonego LP-MP:

$$\mu_{\tilde{\beta}}(\beta) = \begin{cases} \sup\{\mu_{\tilde{\alpha}}(\alpha) : f^+(\alpha) \geq \beta\} & \text{jeśli } \exists \alpha \text{ takie, że } f^+(\alpha) \geq \beta, \\ 0 & \text{w przeciwnym przypadku.} \end{cases} \quad (3.28)$$

- dla rozmytego²-przybliżonego LN-MP:

$$\mu_{\tilde{\beta}}(\beta) = \begin{cases} \sup\{\mu_{\tilde{\alpha}}(\alpha) : f^-(\alpha) \geq \beta\} & \text{jeśli } \exists \alpha \text{ takie, że } f^-(\alpha) \geq \beta, \\ 0 & \text{w przeciwnym przypadku.} \end{cases} \quad (3.29)$$

- dla rozmytego²-przybliżonego UP-MP:

$$\mu_{\tilde{\beta}}(\beta) = \begin{cases} \sup\{\mu_{\tilde{\alpha}}(\alpha) : g^+(\alpha) \leq \beta\} & \text{jeśli } \exists \alpha \text{ takie, że } g^+(\alpha) \leq \beta, \\ 0 & \text{w przeciwnym przypadku.} \end{cases} \quad (3.30)$$

- dla rozmytego²-przybliżonego LP-MP:

$$\mu_{\tilde{\beta}}(\beta) = \begin{cases} \sup\{\mu_{\tilde{\alpha}}(\alpha) : g^-(\alpha) \leq \beta\} & \text{jeśli } \exists \alpha \text{ takie, że } g^-(\alpha) \leq \beta, \\ 0 & \text{w przeciwnym przypadku.} \end{cases} \quad (3.31)$$

Powyższe schematy pozwalają na wnioskowanie z rozmytych przesłanek, nie korzystając w ogóle z typowych rozmytych łączników (t-norm, s-norm). Zapobiega to problemowi wyboru odpowiednich łączników (które mogą być zdefiniowane dość dowolnie), ale przede wszystkim pozwala na wykorzystanie wyłącznie własności porządkowych funkcji przynależności, jest więc zgodne z podstawowym założeniem dotyczącym reguł gradualnych.

3.4 Przykładowy zbiór danych

Rozważania przeprowadzone w tym rozdziale zostaną teraz przedstawione na prostym przykładzie dydaktycznym. Z uwagi na przejrzystość, ograniczymy się do dwóch atrybutów, co w zupełności wystarczy, aby zauważyć, jak działają opisane podejście. Co więcej, nie będziemy przekształcali rzeczywistego zbioru danych, ale rozpoczniemy z punktu, w którym mamy już dane wartości wiarygodności na kryteriach i decyzji. Dane w postaci dwunastu obiektów przedstawia tabela 3.1, przy czym oba kryteria są typu *zysk*. Rozkład obiektów na przestrzeni kryteriów przedstawia rysunek 3.2.

W tabeli 3.1 zostały również wyznaczone górne i dolne przybliżenia zgodnie ze sposobem opisanym w 3.2.4. Aby wyznaczyć wiarygodność przynależności obiektu x do dolnego przybliżenia, szukamy wszystkich obiektów dominujących x , a następnie wybieramy najmniejszą spośród wiarygodności decyzji. Z kolei przy wyznaczaniu wiarygodności przynależności do dolnego przybliżenia, szukamy obiektów zdominowanych przez x i wybieramy największą z wartości.

Same funkcje gradualne, $f(\cdot)$ oraz $g(\cdot)$, wyznaczane są zgodnie z zasadą maksymalnej ostrożności. Można by próbować przedstawić je na dwa sposoby. Jednym z nich jest ujęcie takiej funkcji we wzór z klamrą, podobnie jak w matematyce przedstawia się funkcje zmienne przedziałami:

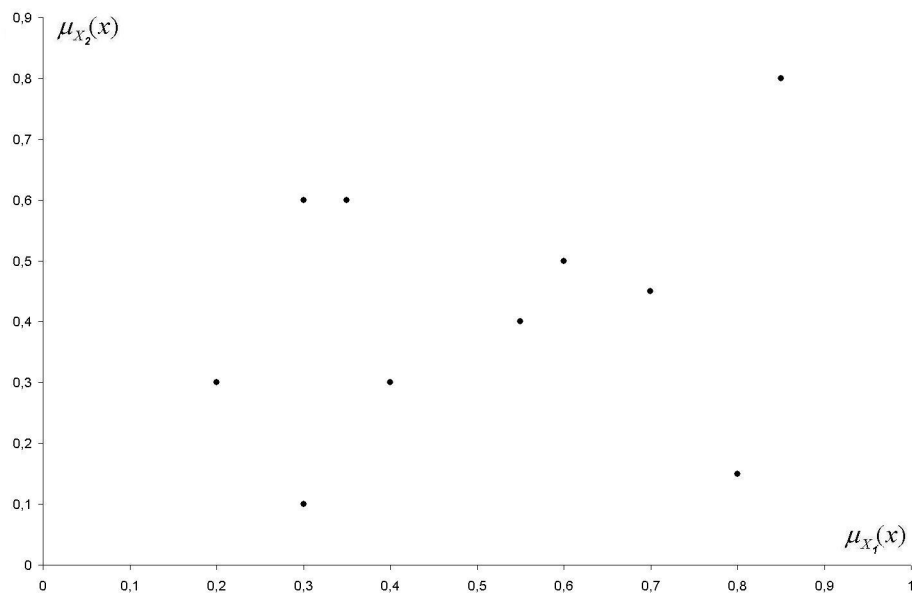
Nr	μ_{X_1}	μ_{X_2}	μ_Y	$\mu_{\underline{App}(X_1, X_2, Y)}$	$\mu_{\overline{App}(X_1, X_2, Y)}$
1	0.3	0.1	0.1	0.1	0.1
2	0.4	0.3	0.2	0.2	0.25
3	0.2	0.3	0.25	0.2	0.25
4	0.6	0.5	0.3	0.3	0.7
5	0.8	0.15	0.5	0.5	0.5
6	0.7	0.45	0.6	0.6	0.7
7	0.3	0.6	0.65	0.55	0.65
8	0.85	0.8	0.75	0.75	0.75
9	0.35	0.6	0.55	0.55	0.65
10	0.55	0.4	0.7	0.3	0.7

Tablica 3.1: Przykładowy zbiór danych. Wartości wiarygodności X_1 , X_2 , decyzji Y oraz dolnego i górnego przybliżenia.

$$f(\alpha) = \begin{cases} 0.75 & \alpha \in [0.85, 1] \times [0.8, 1] \\ 0.6 & \alpha \in [0.7, 1] \times [0.45, 8) \text{ lub } \alpha \in [0.7, 0.85) \times [0.45, 8) \\ 0.55 & \alpha \in [0.3, 7) \times [0.6, 1] \\ 0.5 & \alpha \in [0.8, 1] \times [0.15, 0.45) \\ 0.3 & \alpha \in [0.55, 0.7) \times [0.4, 0.6) \text{ lub } \alpha \in [0.7, 0.8) \times [0.4, 0.45) \\ 0.2 & \alpha \in [0.2, 0.8) \times [0.3, 0.4) \text{ lub } \alpha \in [0.2, 0.55) \times [0.4, 0.6) \\ & \text{lub } \alpha \in [0.2, 0.3) \times [0.6, 1) \\ 0.1 & \alpha \in [0.3, 1] \times [0.1, 0.15) \text{ lub } \alpha \in [0.3, 0.8) \times [0.1, 0.3) \\ 0 & \alpha \in [0, 1] \times [0, 0.1) \text{ lub } \alpha \in [0, 0.3) \times [0.1, 0.3) \\ & \text{lub } \alpha \in [0, 0.2) \times [0.3, 1] \end{cases} \quad (3.32)$$

$$g(\alpha) = \begin{cases} 1 & \alpha \in (0.85, 1] \times [0, 1] \text{ lub } \alpha \in [0, 1] \times (0.8, 1] \\ 0.75 & \alpha \in [0, 0.85] \times (0.6, 0.8] \text{ lub } \alpha \in (0.35, 0.85] \times (0.5, 0.6] \\ & \text{lub } \alpha \in (0.6, 0.8] \times (0.45, 0.5] \text{ lub } \alpha \in (0.7, 0.85] \times (0.15, 0.45] \\ & \text{lub } \alpha \in (0.8, 0.85] \times [0, 0.15] \\ 0.7 & \alpha \in [0, 0.35] \times (0.5, 0.6] \\ 0.65 & \alpha \in (0.6, 0.8] \times [0, 0.15] \\ 0.5 & \alpha \in (0.6, 0.7] \times (0.15, 0.45] \text{ lub } \alpha \in (0.4, 0.6] \times [0, 0.5] \\ & \text{lub } \alpha \in [0, 0.4] \times (0.3, 0.5] \\ 0.25 & \alpha \in [0, 0.3] \times (0.1, 0.3] \text{ lub } \alpha \in (0.3, 0.4] \times (0.1, 0.3] \\ 0.1 & \alpha \in [0, 0.3] \times [0, 0.1] \end{cases} \quad (3.33)$$

Niestety, sposób ten jest nieprzejrzysty i dla dużej ilości obiektów ilość podzbiórów bardzo szybko rośnie. W związku z tym lepszym pomysłem wydaje się przedstawienie funkcji $f(\cdot)$ i $g(\cdot)$ za pomocą zbioru reguł, opartych na obiektach. Zbiory te wyglądają w następujący sposób:



Rysunek 3.2: Przykładowy zbiór danych. Rozłożenie obiektów w przestrzeni kryteriów.

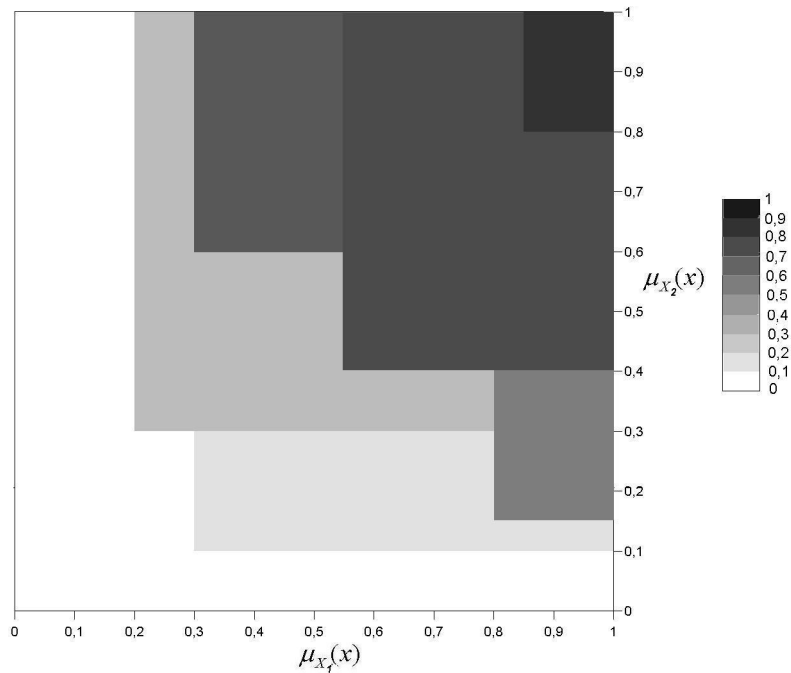
- dla funkcji $f(\cdot)$:

$$\begin{aligned}
 \mu_{X_1}(x) \geq 0.3 \wedge \mu_{X_2}(x) \geq 0.1 &\rightarrow f(x) \geq 0.1 \\
 \mu_{X_1}(x) \geq 0.2 \wedge \mu_{X_2}(x) \geq 0.3 &\rightarrow f(x) \geq 0.2 \\
 \mu_{X_1}(x) \geq 0.55 \wedge \mu_{X_2}(x) \geq 0.4 &\rightarrow f(x) \geq 0.3 \\
 \mu_{X_1}(x) \geq 0.8 \wedge \mu_{X_2}(x) \geq 0.15 &\rightarrow f(x) \geq 0.5 \\
 \mu_{X_1}(x) \geq 0.3 \wedge \mu_{X_2}(x) \geq 0.6 &\rightarrow f(x) \geq 0.55 \\
 \mu_{X_1}(x) \geq 0.7 \wedge \mu_{X_2}(x) \geq 0.45 &\rightarrow f(x) \geq 0.6 \\
 \mu_{X_1}(x) \geq 0.85 \wedge \mu_{X_2}(x) \geq 0.8 &\rightarrow f(x) \geq 0.75
 \end{aligned}$$

- dla funkcji $g(\cdot)$:

$$\begin{aligned}
 \mu_{X_1}(x) \leq 0.85 \wedge \mu_{X_2}(x) \leq 0.8 &\rightarrow g(x) \leq 0.75 \\
 \mu_{X_1}(x) \leq 0.6 \wedge \mu_{X_2}(x) \leq 0.5 &\rightarrow g(x) \leq 0.7 \\
 \mu_{X_1}(x) \leq 0.7 \wedge \mu_{X_2}(x) \leq 0.45 &\rightarrow g(x) \leq 0.7 \\
 \mu_{X_1}(x) \leq 0.35 \wedge \mu_{X_2}(x) \leq 0.6 &\rightarrow g(x) \leq 0.65 \\
 \mu_{X_1}(x) \leq 0.8 \wedge \mu_{X_2}(x) \leq 0.15 &\rightarrow g(x) \leq 0.5 \\
 \mu_{X_1}(x) \leq 0.4 \wedge \mu_{X_2}(x) \leq 0.3 &\rightarrow g(x) \leq 0.25 \\
 \mu_{X_1}(x) \leq 0.3 \wedge \mu_{X_2}(x) \leq 0.1 &\rightarrow g(x) \leq 0.1
 \end{aligned}$$

Zauważmy, że powyższy zapis jest zgodny z definicją reguł gradualnych w



Rysunek 3.3: Przykładowy zbiór danych. Wartości wiarygodności decyzji (oznaczone odcieniami szarości) w przestrzeni kryteriów.

punkcie 3.1.2. Wartość funkcji w każdym punkcie musi być zgodna ze wszystkimi regułami, czyli jeśli dany punkt spełnia część warunkową reguły, to wartość funkcji w tym punkcie również musi spełniać jej część decyzyjną, stąd w szczególności musi spełniać też część decyzyjną najsilniejszej z pokrywających ją reguł. Z uwagi na zasadę maksymalnej ostrożności bierzemy najmniejszą możliwą wartość zgodną z najsilniejszą z reguł, czyli równą parametrowi β (zgodnie z definicjami z 3.1.2) z części decyzyjnej najsilniejszej z reguł.

Poprawność powyższej reprezentacji wynika z faktu, że zgodnie z zasadą maksymalnej ostrożności, wybieramy w danym punkcie wartość funkcji równą największej spośród wartości wiarygodności dolnych przybliżeń obiektów dominowanych przez ten punkt (dominujących dla $g(\cdot)$), a dominowanie obiektu jest równoważne ze spełnieniem części warunkowej reguły opartej na takim obiekcie.

Taki sposób reprezentacji nie pozwala natychmiast ustalić wartości funkcji w danym punkcie (należy przeglądać zbiór implikacji), ale jest dużo bardziej intuicyjny, zwarty, a także prowadzi w kierunku indukcji reguł decyzyjnych, która zostanie omówiona w następnym rozdziale.

Rozdział 4

Indukcja reguł

4.1 Wprowadzenie

Opis danych przez reguły gradualne wykazuje wiele analogii z podejściem do zbiorów rozmytych opartym na relacji dominacji (DRSA). Podobna jest składnia reguł, a poszczególne pojęcia mają swoje odpowiedniki, np. przybliżenia kumulacji klas koncepcyjnie odpowiadają funkcjom $f(\cdot)$ i $g(\cdot)$. Różnice wynikają przede wszystkim z faktu, że wartości decyzji są reprezentowane jako liczby. W ten sposób wiarygodność decyzji może przyjąć dowolną wartość, co powoduje, że liczba takich wartości może być nawet porównywalna z liczbą przykładów (w skrajnym przypadku, jej równa) — z matematycznego punktu widzenia szansa, że dwie, losowo wybrane liczby rzeczywiste powtórzą się, jest zerowa. Sytuację polepsza fakt, że przy zastosowaniu systemu do wspomaganie decyzji, decydent nie jest w stanie podać tak wielu wartości, z uwagi na trudność szacowania wiarygodności decyzji w dokładny sposób przy wyrażaniu swoich preferencji oraz dlatego, że małe różnice wiarygodności nie mają rzeczywistego znaczenia. Mimo to wartości zwykle będzie stosunkowo dużo, szczególnie gdy problem zostałby zastosowany na przykład w sterowaniu, a wyniki byłyby danymi otrzymanymi z przyrządów o dużej rozdzielczości, lub gdyby decydent ułożył dane z tablicy decyzyjnej w ranking (porządek), zamiast szacować ich wiarygodność. Ponieważ poszczególne wartości decyzji koncepcyjnie odpowiadają różnym klasom decyzyjnym, pojawiającym się w DRSA, mamy do czynienia z przypadkiem bardzo wielu klas, co naturalnie zwiększy trudności z jakimkolwiek zastosowanym językiem opisu.

Mimo to, kontynuując rozwijanie analogii między oboma podejściami, nasuwa się pomysł, aby opis zbioru za pomocą funkcji $f(\cdot)$ i $g(\cdot)$ uogólnić. W przypadku DRSA, następowała później indukcja reguł decyzyjnych, pozwalająca na utworzenie hipotezy dotyczącej zbioru w postaci bardziej ogólnej i prostszej. Wiadomo bowiem, że opis w postaci dolnych i górnych przybliżeń kumulacji klas jest nadmiarowy, i mniej zrozumiały od opisu regułowego.

W przypadku reguł gradualnych problem można zdefiniować wykorzystując

warunki *poprawności* (punkt 3.2.3). Mianowicie, aby zachować spójność reguł ze zbiorem danych, uogólnienie musi dawać reguły poprawne, ale już niekoniecznie ostrożne. Rezygnacja z zasady maksymalnej ostrożności może prowadzić do reguł potencjalnie prostszych, a równocześnie nadal poprawnie opisujących objekty z tablicy decyzyjnej. Proponujemy przeprowadzenie takiego uogólnienia na dwa sposoby: za pomocą redukcji atrybutów oraz poprzez zastosowanie algorytmu indukcji reguł decyzyjnych.

4.2 Uogólnianie poprzez redukcję atrybutów

Aby rozważania stały się bardziej przejrzyste, uprościmy notację definiując zbiór $\mathcal{X} = \{X_{i_1}^\uparrow, \dots, X_{i_r}^\uparrow, X_{j_1}^\downarrow, \dots, X_{j_s}^\downarrow\}$, oraz zbiory $\mathcal{X}^\uparrow = \{X_{i_1}^\uparrow, \dots, X_{i_r}^\uparrow\}$ i $\mathcal{X}^\downarrow = \{X_{j_1}^\downarrow, \dots, X_{j_s}^\downarrow\}$. W podobny sposób będziemy definiowali również każdy podzbiór atrybutów ze zbioru \mathcal{X} . Tym samym dolne i górne przybliżenie można zapisać jako $\underline{App}(\mathcal{X}, Y)$ i $\overline{App}(\mathcal{X}, Y)$.

4.2.1 Jakość klasyfikacji i redukty

Analogicznie do DRSA, jesteśmy w stanie zdefiniować współczynnik *jakości klasyfikacji*. Dla danego zbioru atrybutów \mathcal{X}' może wyznaczyć dla każdego wariantu $x \in U$ wiarygodność przynależności do dolnego i górnego przybliżenia. W przypadku DRSA jakość klasyfikacji określał stosunek obiektów jednoznacznie sklasyfikowanych (nie należących do brzegu żadnej z klas) do wszystkich obiektów z U . W przypadku reguł gradualnych obiekt $x \in U$ sklasyfikowany jednoznacznie na decyzji Y w opraciu o wiedzę o \mathcal{X}' , to taki obiekt, dla którego zachodzi:

$$\mu[\underline{App}(\mathcal{X}', Y), x] = \mu[\overline{App}(\mathcal{X}', Y), x] \quad (4.1)$$

Stąd współczynnik *jakości klasyfikacji* definiujemy jako:

$$\gamma(\mathcal{X}, Y) = \frac{|\{x \in U : \mu[\underline{App}(\mathcal{X}, Y), x] = \mu[\overline{App}(\mathcal{X}, Y), x]\}|}{|U|} \quad (4.2)$$

Współczynnik $\gamma(\mathcal{X}, Y)$ jest więc miarą niespójności zbioru U .

Redukt, na potrzeby następnego podrozdziału, zdefiniujemy nieco inaczej, przyjmując, że jest to taki podzbiór atrybutów $\mathcal{X}' \subseteq \mathcal{X}$, dla którego zachodzi dla każdego obiektu $x \in U$:

$$\mu[\underline{App}(\mathcal{X}', Y), x] = \mu[\underline{App}(\mathcal{X}, Y), x] \quad (4.3)$$

$$\mu[\overline{App}(\mathcal{X}', Y), x] = \mu[\overline{App}(\mathcal{X}, Y), x] \quad (4.4)$$

oraz jest minimalny ze względu na zawieranie. Innymi słowy, zastosowanie reduktu, zamiast pełnego zbioru kryteriów, nie zmienia górnych i dolnych przybliżeń żadnego z obiektów.

Zauważmy, że definicja ta pociąga za sobą własność, która w poprzednich rozdziałach definiowała redukt: dla dowolnego reduktu $\mathcal{X}' \subseteq \mathcal{X}$ zachodzi $\gamma(\mathcal{X}', Y) = \gamma(\mathcal{X}, Y)$. Implikacja w odwrotną stronę niestety nie zachodzi.

Zdefiniujemy również redukt lokalny dla podejścia opartego na regułach gradualnych w następujący sposób. Reduktem lokalnym dominującym $r^+(x) \subseteq \mathcal{X}$ dla obiektu $x \in U$ nazywamy pozbior kryteriów spełniający warunek:

$$\forall y \in U: \mu_Y(y) < \mu_Y(x) \rightarrow y \notin D_{r^+(x)}^\uparrow(x) \quad (4.5)$$

oraz minimalny ze względu na zawierania. Przyjmując, że $r^+(x) = \{X_{i_1}^\uparrow, \dots, X_{i_m}^\uparrow, X_{j_1}^\downarrow, \dots, X_{j_n}^\downarrow\}$, zbiór $D_{r^+(x)}^\uparrow(x)$ definiujemy jako:

$$D_{r^+(x)}^\uparrow(x) = \{z \in U: \mu_{X_h}(z) \geq \mu_{X_h}(x) \forall X_h \in \{X_{i_1}^\uparrow, \dots, X_{i_m}^\uparrow\} \wedge \wedge \mu_{X_h}(z) \leq \mu_{X_h}(x) \forall X_h \in \{X_{j_1}^\downarrow, \dots, X_{j_n}^\downarrow\}\}. \quad (4.6)$$

Podobnie, redukt lokalnym zdominowanym $r^-(x) \subseteq \mathcal{X}$ dla obiektu $x \in U$ nazywamy pozbior kryteriów spełniający warunek:

$$\forall y \in U: \mu_Y(y) > \mu_Y(x) \rightarrow y \notin D_{r^-(x)}^\downarrow(x) \quad (4.7)$$

oraz minimalny ze względu na zawierania. Przyjmując, że $r^-(x) = \{X_{i_1}^\uparrow, \dots, X_{i_v}^\uparrow, X_{j_1}^\downarrow, \dots, X_{j_w}^\downarrow\}$, zbiór $D_{r^-(x)}^\downarrow(x)$ definiujemy jako:

$$D_{r^-(x)}^\downarrow(x) = \{z \in U: \mu_{X_h}(z) \leq \mu_{X_h}(x) \forall X_h \in \{X_{i_1}^\uparrow, \dots, X_{i_v}^\uparrow\} \wedge \wedge \mu_{X_h}(z) \geq \mu_{X_h}(x) \forall X_h \in \{X_{j_1}^\downarrow, \dots, X_{j_w}^\downarrow\}\}. \quad (4.8)$$

4.2.2 Redukcja atrybutów

Uogólnienie reguł poprzez redukcję atrybutów polega na zastosowaniu zasady maksymalnej ostrożności, ale nie do pełnego zbioru \mathcal{X} , lecz do jednego z reduktów $\mathcal{X}' \subseteq \mathcal{X}$. Pokażemy, że w takim przypadku mimo, że reguła będzie maksymalnie ostrożna względem \mathcal{X}' , utraci tę właściwość względem \mathcal{X} , równocześnie zachowując *poprawność*.

Zauważmy, że wykorzystując prostszą notację, zbiory $D^\downarrow(x)$ i $D^\uparrow(x)$ można zdefiniować następująco:

$$D^\uparrow(x) = \{z \in U: \mu_{X_h}(z) \geq \mu_{X_h}(x) \forall X_h \in \mathcal{X}^\uparrow \wedge \wedge \mu_{X_h}(z) \leq \mu_{X_h}(x) \forall X_h \in \mathcal{X}^\downarrow\}. \quad (4.9)$$

$$D^\downarrow(x) = \{z \in U: \mu_{X_h}(z) \leq \mu_{X_h}(x) \forall X_h \in \mathcal{X}^\uparrow \wedge \wedge \mu_{X_h}(z) \geq \mu_{X_h}(x) \forall X_h \in \mathcal{X}^\downarrow\}. \quad (4.10)$$

Podobnie zdefiniujemy zbiory $D'^{\downarrow}(x)$ i $D'^{\uparrow}(x)$ oparte o rodzinę \mathcal{X}' :

$$D'^{\uparrow}(x) = \{z \in U : \mu_{X_h}(z) \geq \mu_{X_h}(x) \forall X_h \in \mathcal{X}'^{\uparrow} \wedge \wedge \mu_{X_h}(z) \leq \mu_{X_h}(x) \forall X_h \in \mathcal{X}'^{\downarrow}\}. \quad (4.11)$$

$$D'^{\downarrow}(x) = \{z \in U : \mu_{X_h}(z) \leq \mu_{X_h}(x) \forall X_h \in \mathcal{X}'^{\downarrow} \wedge \wedge \mu_{X_h}(z) \geq \mu_{X_h}(x) \forall X_h \in \mathcal{X}'^{\uparrow}\}. \quad (4.12)$$

Zauważmy, że ponieważ $\mathcal{X}' \subseteq \mathcal{X}$, to przynależność danego obiektu $x \in U$ do zbioru $D^{\downarrow}(x)$ (lub $D^{\uparrow}(x)$) pociąga za sobą (z definicji tych zbiorów) przynależność do $D'^{\downarrow}(x)$ (lub $D'^{\uparrow}(x)$), co oznacza, że:

$$D^{\downarrow}(x) \subseteq D'^{\downarrow}(x) \quad (4.13)$$

$$D^{\uparrow}(x) \subseteq D'^{\uparrow}(x) \quad (4.14)$$

Z definicji reduktu wynika, że wiarygodności dolnych i górnych przybliżenia nie zmieniły się. Stąd, funkcje $f'(\cdot)$ i $g'(\cdot)$ wyznaczone na podstawie \mathcal{X}' będą poprawne (po rozszerzeniu dziedziny funkcji ze zbioru \mathcal{X}' na zbiór \mathcal{X}) względem \mathcal{X} . Co więcej, stosując zasadę maksymalnej ostrożności względem podzbioru \mathcal{X}' , czyli definiując funkcje zgodnie z (3.16) i (3.17), z powodu związków (4.13) i (4.14), zachodzi dla dowolnych $\alpha_{i_1}, \dots, \alpha_{i_r}, \alpha_{j_1}, \dots, \alpha_{j_s}$:

$$f'(\alpha_{i_1}, \dots, \alpha_{i_{r'}}, \alpha_{j_1}, \dots, \alpha_{j_{s'}}) \geq f(\alpha_{i_1}, \dots, \alpha_{i_r}, \alpha_{j_1}, \dots, \alpha_{j_s}) \quad (4.15)$$

$$g'(\alpha_{i_1}, \dots, \alpha_{i_{r'}}, \alpha_{j_1}, \dots, \alpha_{j_{s'}}) \leq g(\alpha_{i_1}, \dots, \alpha_{i_r}, \alpha_{j_1}, \dots, \alpha_{j_s}) \quad (4.16)$$

gdzie $r' = |\mathcal{X}'^{\uparrow}|$, $s' = |\mathcal{X}'^{\downarrow}|$, $r = |\mathcal{X}^{\uparrow}|$ i $s = |\mathcal{X}^{\downarrow}|$. Innymi słowy, maksymalnie ostrożne funkcje $f'(\cdot)$ i $g'(\cdot)$ przestają być maksymalnie ostrożne na zbiorze \mathcal{X} (po rozszerzeniu dziedziny), tzn. stają się funkcjami ogólniejszymi. Są równocześnie prostsze z uwagi na mniejszą liczbę argumentów.

4.3 Indukcja reguł decyzyjnych z macierzy dominacji

Metoda oparta na reduktach daje uproszczony i bardziej ogólny opis, z uwagi na lepsze szacowanie wartości wiarygodności w punktach przestrzeni, gdzie nie znajdują się obiekty, a przede wszystkim zmniejsza liczbę kryteriów, czyniąc problem bardziej przejrzystym. Poza tym pozostawia opis zbioru zgodny ze składnią funkcji gradualnych. Niestety, daje się zastosować tylko w przypadku, gdy pewne kryteria są wyraźnie nadmiarowe — wyłącznie wtedy pomimo wyboru tylko części z nich (projekcji przestrzeni kryteriów na podprzestrzeń) nie zmieniają się wiarygodności dolnych i górnych przybliżeń dla żadnych obiektów.

Co więcej, nie daje na tyle czytelnych i prostych hipotez, jak w przypadku użycia algorytmu indukcji reguł. W związku z tym przedstawione zostanie inne podejście, oparte na pozyskiwaniu reguł gradualnych z macierzy dominacji na podstawie reduktów lokalnych. Z uwagi na przejrzystość opisu, nie będziemy przedstawiali reguł w postaci funkcji $f(\cdot)$ i $g(\cdot)$ — tak, jak robiliśmy to dotychczas. Zamiast tego reguły będą reprezentowane przez pojedyncze implikacje, a nie — jak dotychczas — przez zbiory implikacji o części decyzyjnej powiązanej zależnością funkcyjną.

Istnieje szereg algorytmów indukcji reguł decyzyjnych, problem ten doczekał się dość znacznej ilości rozwiązań, ponieważ stanowi jeden z ważniejszych aspektów maszynowego uczenia się. Prawdopodobnie, po niewielkich przekształceniach, większość z nich dałoby się zastosować do rozważanego problemu, z uwagi na fakt, że reguły gradualne pasują do ogólnego wzoru reguły: części warunkowej składającej się z koniunkcji selektorów oraz części decyzyjnej oznaczającej przydział do klasy. Przykładowo, z powodzeniem zastosowany został do indukcji reguł w DRSA algorytm DOMLem [31]. W przypadku podejścia z regułami gradualnymi problemem jest jednak duża liczba klas - jeśli uznać za klasy, poprzez analogię do DRSA, wartości wiarygodności decyzji obiektów. Stąd zdecydowano się na algorytm, który generuje reguły nie w oparciu o pokrywanie klas, ale o dyskryminację pojedynczych przykładów. Algorytm spełniający to kryterium, ADRIA (*All-Dominance-based-Rules Induction Algorithm*) został przedstawiony przez K. Dembczyńskiego, R. Pindura i R. Susmagę [6]. Opiera się na pojęciu macierzy dominacji, generacji reduktów lokalnych, na tej podstawie utworzeniu zbioru reguł bazujących na obiektach a następnie selekcji reguł minimalnych, w wyniku czego utworzony zostaje wyczerpujący zbiór reguł opisujący dane z tablicy decyzyjnej.

4.3.1 Generowanie reduktów lokalnych

Punktem wyjścia dla algorytmu generowania reduktów jest utworzenia macierzy dominacji $\mathcal{D}(\mathcal{P})$ na podstawie zbioru kryteriów \mathcal{P} . Przypomnijmy, że zbiór $\delta^{\succeq}(x, y)$ macierzy $\mathcal{D}(\mathcal{P})$ zawiera wszystkie kryteria, na których obiekt x przewyższa obiekt y . W przypadku, gdy $\delta^{\succeq}(x, y) = \mathcal{P}$, to obiekt x dominuje obiekt y , natomiast gdy $\delta^{\succeq}(x, y) = \delta^{\succeq}(y, x) = \mathcal{P}$, to obiekty są nierozróżnialne.

Zdefiniujemy następnie funkcję dominacji $Df(\mathcal{P})$ na zbiorze kryteriów \mathcal{P} w następujący sposób [6]. Każdemu kryterium $q \in \mathcal{P}$ przypiszemy zmienną boolowską \bar{q} i niech $\bigcup \neg \delta^{\succeq}(x, y)$ oznacza sumę boolowską wszystkich zanegowanych zmiennych boolowskich przypisanych do zbioru kryteriów $\delta^{\succeq}(x, y)$.

Funkcją \mathcal{P} -dominacji dla lokalnego reduktu dominującego, bazującą na obiekcie x o wiarygodności dolnego przybliżenia $t = \mu[\underline{App}(\mathcal{P}, Y), x]$ oznaczamy przez $Df_t^{x \succcurlyeq}$ i definiujemy jako:

$$Df_t^{x \succcurlyeq}(\mathcal{P}) = \prod \left\{ \bigcup \neg \delta^{\succeq}(y, x) : \mu[\overline{App}(\mathcal{P}, Y), y] < t \right\} \quad (4.17)$$

Funkcją \mathcal{P} -dominacji dla lokalnego reduktu zdominowanego, bazującą na

obiekcie x o wiarygodności decyzji $s = \mu_Y(x)$ oznaczamy przez $Df^{x \leq}$ i definiujemy jako:

$$Df_s^{x \leq}(\mathcal{P}) = \prod \left\{ \bigcup \neg \delta^{\leq}(y, x) : \mu[\overline{App}(\mathcal{P}, Y), y] > s \right\} \quad (4.18)$$

Przekształcenie tych funkcji od postaci koniunkcyjnej do nieredundantnej postaci dysjunkcyjnej odpowiada problemowi znajdowania wszystkich reduktów lokalnych bazujących na obiekcie x . Postać dysjunkcyjna to dysjunkcja koniunkcji, przy czym każda z koniunkcji opisują jeden z reduktów lokalnych dominujących (dla funkcji $Df^{x \geq}$) lub zdominowanych (dla funkcji $Df^{x \leq}$).

Powyższe przekształcenie może być wykonane za pomocą *szybkiego algorytmu generowania reduktów* (SAGR) opisanego przez R. Susmagę [33, 34]. W wyniku jego działania dla danego obiektu $x \in U$ otrzymujemy zbiór lokalnych reduktów dominujących i zdominowanych, opartych na tym obiekcie.

4.3.2 Tworzenie reguł

Na podstawie dominujących i zdominowanych reduktów lokalnych w sposób bezpośredni można utworzyć reguły decyzyjne. Rozważmy redukt dominujący $r^+(x) = \{X_{i_1}^\uparrow, \dots, X_{i_m}^\uparrow, X_{j_1}^\downarrow, \dots, X_{j_n}^\downarrow\}$, wygenerowany za pomocą algorytmu SAGR, oparty na obiekcie $x \in U$. Przyjmijmy również, że $\mu[\underline{App}(\mathcal{X}, Y), x] = \beta$ i dla każdego $X_{i_k}^\uparrow$, $\mu_{X_{i_k}^\uparrow}(x) = \alpha_{i_k}$ oraz dla każdego $X_{j_k}^\downarrow$, $\mu_{X_{j_k}^\downarrow}(x) = \alpha_{j_k}$. W takim przypadku na podsacie znajomości reduktu można utworzyć regułę o następującej postaci:

$$\begin{aligned} \text{jeśli } \mu_{X_{i_1}^\uparrow}(x) \geq \alpha_{i_1} \wedge \dots \wedge \mu_{X_{i_m}^\uparrow}(x) \geq \alpha_{i_m} \wedge \mu_{X_{j_1}^\downarrow}(x) \leq \alpha_{j_1} \wedge \dots \wedge \\ \wedge \mu_{X_{j_m}^\downarrow}(x) \leq \alpha_{j_m} \text{ to } \mu_Y(x) \geq \beta \end{aligned} \quad (4.19)$$

O regule takiej mówimy, że *bazuje na obiekcie x* . Nazwiemy ją, analogicznie do DRSA, D^{\geq} -regułą gradualną. Zauważmy, że reguła taka jest spójna z definicją podaną w 3.1.2, przy czym stwierdzenia typu „ $x \in X_{i_k}^\uparrow$ z wiarygodnością $C(X_{i_k}^\uparrow) \geq \alpha_{i_k}$ ” należy traktować jako równoważne stwierdzeniom „ $\mu_{X_{i_k}^\uparrow}(x) \geq \alpha_{i_k}$ ”.

Pokażemy teraz, że reguła taka jest poprawna zgodnie z definicją z 3.2.3 (dowód jest intuicyjnie bardzo łatwy, problem pojawia się przy jego sformalizowaniu). Weźmy rozważany dotychczas obiekt x , na którym zbudowana jest reguła oraz dowolny punkt w pełnej przestrzeni kryteriów $\alpha \in [0, 1]^{r+s}$ taki, że $x \in D^\downarrow(\alpha)$. Ponieważ zachodzi $D^\downarrow(\alpha) \subseteq D_{r^+(x)}^\downarrow(\alpha)$ (zmniejsza się liczba warunków do spełnienia, aby obiekt przynależał do zbioru $D_{r^+(x)}^\downarrow(\alpha)$, stąd $D_{r^+(x)}^\downarrow(\alpha)$ staje się obszerniejszy od $D^\downarrow(\alpha)$), więc $x \in D_{r^+(x)}^\downarrow(\alpha)$. Dalej, ponieważ w zbiorze $D_{r^+(x)}^\downarrow(\alpha)$ znajdują się takie obiekty z , że $\mu_{X_h}(z) \leq \mu_{X_h}(\alpha_{i_h})$ dla wszystkich $X_h \in \{X_{i_1}^\uparrow, \dots, X_{i_m}^\uparrow\}$ oraz $\mu_{X_h}(z) \geq \mu_{X_h}(\alpha_{j_h})$ dla wszystkich $X_h \in \{X_{j_1}^\downarrow, \dots, X_{j_n}^\downarrow\}$, stąd α jest pokrywany przez rozważaną wyżej

regułę opartą na obiekcie x (spełnia jej część warunkową), co oznacza, że $\mu_Y(\alpha) \geq \beta = \mu[\underline{App}(X_{i_1}^\uparrow, \dots, X_{i_r}^\uparrow, X_{j_1}^\downarrow, \dots, X_{j_s}^\downarrow, Y), x]$.

Tym samym pokazaliśmy, że reguła jest poprawna. Oczywiście, warunki poprawności powinny zachodzić dla każdego obiektu $x \in U$, a wykazaliśmy ich spełnienie wyłącznie przez obiekt, na którym oparta jest reguła. Jednak algorytm generuje redukty lokalne na każdym obiekcie $x \in U$, a następnie na każdym obiekcie tworzy z nich reguły podobne do wyżej opisanej (z których każda jest poprawna). Stąd bardziej precyzyjnie należy powiedzieć, że dopiero zbiór reguł spełnia warunki poprawności. W dalszej części rozdziału pokażemy, że minimalizacja zbioru reguł zachowuje poprawność.

Podobnie, rozważając redukt zdominowany oparty na obiekcie $x \in U$, $r^-(x) = \{X_{i_1}^\uparrow, \dots, X_{i_m}^\uparrow, X_{j_1}^\downarrow, \dots, X_{j_n}^\downarrow\}$, oraz przyjmując, że $\mu[\overline{App}(\mathcal{X}, Y), x] = \beta$, dla każdego $X_{i_k}^\uparrow$, $\mu_{X_{i_k}^\uparrow}(x) = \alpha_{i_k}$ i dla każdego $X_{j_k}^\downarrow$, $\mu_{X_{j_k}^\downarrow}(x) = \alpha_{j_k}$ można utworzyć regułę o następującej postaci:

$$\begin{aligned} \text{jeśli } \mu_{X_{i_1}^\uparrow}(x) \leq \alpha_{i_1} \wedge \dots \wedge \mu_{X_{i_m}^\uparrow}(x) \leq \alpha_{i_m} \wedge \mu_{X_{j_1}^\downarrow}(x) \geq \alpha_{j_1} \wedge \dots \wedge \\ \wedge \mu_{X_{j_m}^\downarrow}(x) \geq \alpha_{j_m} \text{ to } \mu_Y(x) \leq \beta \end{aligned} \quad (4.20)$$

Regułę taką nazwiemy D^{\leq} -regułą gradualną. Poprawność reguły można pokazać analogicznie, jak w przypadku poprzednim.

Zwróćmy również uwagę, że w wyniku działania algorytmu mogą zostać wygenerowane reguły o części warunkowej takiej jak (4.19) lub (4.20), ale słabszej części decyzyjnej. Przykładowo, zamiast konkluzji w postaci „ $\mu_Y(x) \geq \beta$ ”, można przyjąć „ $\mu_Y(x) \geq \beta'$ ”, gdzie $\beta' \leq \beta$, o ile istnieje obiekt $y \in U$ taki, że $\mu[\underline{App}(\mathcal{X}, Y), y] = \beta'$ (istnienie obiektu wynika z faktu, że nie posługujemy się żadnymi wartościami wiarygodności, które nie wystąpiły w tablicy decyzyjnej, z uwagi na wykorzystywanie tylko cech porządkowych). Wydaje się, że taka reguła nie spełnia już warunków poprawności z uwagi na to, że ma za słabą konkluzję, ale zwróćmy uwagę, że przy wyznaczaniu wartości wiarygodności decyzji w danym punkcie za pomocą zbioru reguł, bierzemy najsilniejsze oszacowanie. Stąd wystarczy, że będzie w zbiorze reguła oparta na tym samym przykładzie, lub pokrywająca ten przykład, o części decyzyjnej „ $\mu_Y(x) \geq \beta$ ”. Okazuje się, że taka reguła zawsze musi się w zbiorze pojawić (inaczej przeczyłoby to bowiem warunku minimalności), co pokażemy w dalszej części rozdziału.

Ten sposób tworzenia reguły r_l na podstawie obiektu x_i w oparciu o redukt $r^\pm(x_i)$, na potrzeby algorytmu można uznać za funkcję *Trace* (z ang. *ślada*), którą zapiszemy w następujący sposób: $r_l = \text{Trace}(r^\pm(x_i), x_i)$.

4.3.3 Algorytm indukcji

Algorytm indukcji reguł z macierzy dominacji można zapisać zgodnie z tablicą 4.1. Generuje on wyczerpujący zbiór reguł (zbiór wszystkich możliwych reguł) minimalnych. Aby utworzyć reguły o konkluzji „ $\mu_Y(x) \geq \beta$ ” (przy czym β musi być wiarygodnością dolnego przybliżenia decyzji jakiegoś obiektu, a

Wejście: Zbiór przykładów pozytywnych Pos , zbiór przykładów negatywnych Neg

Wyjście: Zbiór wszystkich reguł minimalnych R , który odróżnia obiekty należące do Pos od obiektów należących do Neg .

Indukcja wyczerpującego zbioru reguł minimalnych:

- Krok 1:** $R = \emptyset$
Krok 2: dla każdego $i = 1 \dots |Pos|$
Krok 3: $R(x_i = SAGR(x_i, Neg);$
Krok 4: dla każdego $j = 1 \dots |R(x_i)|$
Krok 5: $r_l = Trace(r_j, x_i); r_j \in R(x_i)$
Krok 6: $Min(r_l, x_i, R;$
Krok 7: jeśli r_l jest minimalna, to $R = R \cup r_l;$

Tablica 4.1: Schemat algorytmu generującego wyczerpujący zbiór reguł.

nie dowolną liczbą), wybieramy zbiór wszystkich obiektów $x \in U$, dla których $\mu_Y(x) \geq \beta$ dla D^{\geq} -reguły gradualnej (lub $\mu_Y(x) \leq \beta$ dla D^{\leq} -reguły), czyli spełniający części decyzyjne tworzonych reguł. Zbiór taki nazwiemy *zbiorem przykładów pozytywnych* i oznaczymy przez Pos , natomiast zbiór wszystkich pozostałych obiektów nazwiemy *zbiorem przykładów negatywnych* i oznaczymy przez Neg . Następnie za pomocą algorytmu SAGR, dla każdego obiektu $x \in Pos$ generujemy zbiór reduktów i na jego podstawie zbiór reguł (funkcja *Trace*). Regułę dołącza się do tworzonych reguł tylko wtedy, gdy przejdzie pomyślnie test *Min*, co oznacza że jest minimalna.

Reguła jest minimalna, gdy nie istnieje inna reguła z częścią warunkową co najmniej tak samo słabą (ogólną) i z częścią decyzyjną co najmniej tak samo silną. Mówimy, że reguła r_i jest *silniejsza* od r_j , jeśli r_i ma słabszą (bardziej ogólną) lub taką samą część warunkową i silniejszą lub taką samą część decyzyjną, niż r_j . Stąd reguła jest minimalna, jeśli nie ma silniejszej reguły. Algorytm, który pozwala na podstawie macierzy dominacji ustalić, czy dana reguła jest minimalna, został opracowany przez K. Dembczyńskiego [6] i nie zostanie opisany z uwagi na brak bezpośredniego związku z tematem niniejszej pracy. W schemacie ogólnym w tablicy 4.1 został oznaczony jako funkcja *Min*.

Odrzucanie reguł nieminimalnych nie wpływa na poprawność opisu, omawianą w poprzednim podrozdziale. Wynika to z faktu, że z każdym obiektem $x \in U$ (którego wiarygodność dolnego (górnego) przybliżenia oznaczymy jako β) związana jest (czyli bazuje na x lub pokrywa x) przynajmniej jednak reguła o części decyzyjnej przynajmniej tak silnej, jak postać: „ $\mu_Y(x) \geq \beta$ ”. Gdyby takiej reguły nie było, moglibyśmy ją dodać do zbioru, byłaby oparta na obiekcie x . Reguła taka byłaby minimalna, gdyż nie było innej reguły która pokrywałaby, lub była oparta na x (o co najmniej tak samo ogólnej części warunkowej),

a zarazem miała co najmniej tak silną część decyzyjną. Tym samym, skoro moglibyśmy dodać jeszcze jedną regułę do zbioru, zbiór reguł nie byłby zbiorem wyczerpującym, co prowadzi do sprzeczności.

Jako wynik działania algorytmu ADRIA, otrzymujemy więc wyczerpujący (pełny) zbiór reguł minimalnych. Zbiór ten jest poprawny w sensie definicji z 3.2.3, mimo to jest nadmiarowy — usunięcie reguł ze zbioru nie musi spowodować utraty poprawności. Warunek, jaki musi spełniać zbiór, aby był poprawny, jest następujący:

- dla D^{\geq} -reguł gradualnych: każdy obiekt $x \in U$ o wiarygodności dolnego przybliżenia $\mu[\underline{App}(\mathcal{X}, Y), x] = \beta$, musi być pokryty przez przynajmniej jedną regułę o części decyzyjnej przynajmniej tak silnej, jak $\mu_Y(x) \geq \beta$ (w szczególności reguła taka może bazować na obiekcie x);
- dla D^{\leq} -reguł gradualnych: każdy obiekt $x \in U$ o wiarygodności górnego przybliżenia $\mu[\overline{App}(\mathcal{X}, Y), x] = \beta$, musi być pokryty przez przynajmniej jedną regułę o części decyzyjnej przynajmniej tak silnej, jak $\mu_Y(x) \leq \beta$ (w szczególności reguła taka może bazować na obiekcie x).

Zbiór wyczerpujący zawiera zazwyczaj wiele reguł, których odrzucenie nie spowoduje naruszenia jakiegokolwiek z tych warunków. Zbiorem minimalnym nazywamy taki zbiór, że przy usunięciu z niego jakiegokolwiek reguły zostanie naruszony jeden z wymienionych warunków, a zarazem jest on najmniejszym zbiorem spełniającym warunki poprawności. Minimalizacja zbioru nie narusza więc jego poprawności, powoduje za to tworzenie coraz mniej ostrożnych reguł. Jest bardzo korzystna z punktu widzenia opisu zbioru, gdyż powoduje, że opis ten staje się zdecydowanie prostszy i bardziej zrozumiały, a nie dzieje się to kosztem jakości klasyfikacji. Niestety, problem znalezienia minimalnego zbioru reguł jest NP-trudny i nie jest znany jakikolwiek algorytm wielomianowy rozwiązujący powyższe zagadnienie. W związku z tym naturalnym pomysłem jest zastosowanie algorytmów przybliżonych, które zostaną opisane w dalszej części rozdziału.

4.3.4 Selekcja reguł na podstawie wsparcia

Aby otrzymać zbiór minimalny, lub przynajmniej dążyć do jego uzyskania, należy dokonać selekcji reguł ze zbioru wyczerpującego, pozostawiając w zbiorze tylko część z nich, resztę odrzucając. W tym podrozdziale przedstawione zostaną dwa podejścia oparte na przyporządkowaniu każdej z reguł pewnych miar, świadczących o ich jakości. Miary te, z uwagi na stosowanie wyłącznie własności porządkowych, mogą zostać zdefiniowane i obliczone wyłącznie na podstawie wiadomości o obiektach. Natępnie stosowany jest algorytm zachłanny, który odrzuca reguły w kolejności malejących wartości miary, aż do czasu, gdy usunięcie kolejnej reguły spowodowałoby naruszenie warunku poprawności, lub który dodaje do zbioru do czasu, aż nie osiągnie się warunku poprawności.

Reguła wspiera obiekt, jeśli obiekt taki spełnia jej część warunkową. Mając regułę r , zapisaną w postaci implikacji jako $\phi \rightarrow \psi$, przez $[r]$ (lub $[\phi \rightarrow \psi]$)

oznaczymy zbiór wszystkich obiektów spełniających zarówno część warunkową, jak i decyzyjną r , przez $[\phi]$ oznaczymy zbiór wszystkich obiektów spełniających część warunkową, natomiast przez $[\psi]$ - część decyzyjną. Aby zbiór reguł był spójny ze zbiorem obiektów, musi zachodzić $[r] = [\phi]$, stąd będziemy przyjmowali, że jeśli reguła wspiera obiekt, to obiekt ten spełnia równocześnie jej część decyzyjną.

Względny wsparciem (WP) reguły r nazywamy miarę zdefiniowaną w następujący sposób:

$$WP(r) = \frac{|[\phi \rightarrow \psi]|}{|[\psi]|} \quad (4.21)$$

WP jest więc stosunkiem liczby obiektów, które reguła wspiera (i spełniają jej część decyzyjną) do liczby obiektów, które spełniają jej część decyzyjną (ale niekoniecznie muszą być przez nią pokryte). Miara ta świadczy o ogólności reguły — jaką część obiektów pokrywa ona spośród wszystkich tych, które mogłyby potencjalnie pokrywać z uwagi na część decyzyjną.

Pełnym wsparciem (PP) D^{\geq} -reguły gradualnej r o części decyzyjnej „ $\mu_Y(x) \geq \beta$ ” nazywamy miarę:

$$PP(r) = |\{x \in U : \mu[\underline{App}(\mathcal{X}, Y), x] = \beta\}| \quad (4.22)$$

lub, w przypadku D^{\leq} -reguły gradualnej r o części decyzyjnej „ $\mu_Y(x) \leq \beta$ ”, miarę:

$$PP(r) = |\{x \in U : \mu[\overline{App}(\mathcal{X}, Y), x] = \beta\}| \quad (4.23)$$

PP wyraża więc ilość przykładów, które reguła pokrywa „w pełni”, czyli tak, że wartość wiarygodności ich dolnego (górnego) przybliżenia spełnia część decyzyjną w sposób równościowy (zamieniając nierówność w równość), na przykład dla reguły o części decyzyjnej „ $\mu_Y(x) \leq \beta$ ” w pełni pokryte są tylko tylko przykłady o wiarygodności dolnego przybliżenia równym dokładnie β (i oczywiście wspierające regułę). Z uwagi na fakt, że aby zbiór reguł pozostał poprawny, każdy przykład musi zostać w ten sposób pokryty przez przynajmniej jedną regułę, suma PP wszystkich reguł w zbiorze nie może być nigdy mniejsza od ilości przykładów. Jest to dość znaczący współczynnik. Niestety, z uwagi na fakt, że atrybut decyzyjny w podejściu opartym na regułach gradualnych przyjmuje wartości ciągle, wiarygodności decyzji dla obiektów powtarzają się bardzo rzadko (a nawet wcale). Powoduje to, że zazwyczaj reguła o danej części decyzyjnej będzie pokrywać „w pełni” zaledwie jeden obiekt, czyli $PP = 1$ dla większości reguł.

Dwa powyższe współczynniki wystarczą już, aby dokonać selekcji reguł w celu minimalizacji zbioru. Wystarczy wybierać (do zbioru) reguły z najwyższym WP i najwyższym PP , przy czym wydaje się, że drugi ze współczynników jest ważniejszy. Można je stosować w sposób leksykograficzny, porównując najpierw pod względem PP , a dopiero w przypadku równych wartości — pod względem WP , przy czym jeśli wybraliśmy już jakieś reguły (regułę) do zbioru, które „w

Wejście: Wyczerpujący zbiór reguł R

Wyjście: Zredukowany zbiór reguł R taki, że usunięcie z niego jakiegokolwiek reguły spowoduje, że nie będą spełnione warunki poprawności

Algorytm selekcji reguł w oparciu o stopień przekrywania:

1. Wyznacz wartość SP dla każdej pary reguł $(r_i, r_j) \in R \times R$, a wyniki zapisz w postaci macierzy $\mathcal{SP} = \{SP(r_i, r_j) : r_i, r_j \in R\}$
2. Ze zbioru R wybierz parę reguł (r_i, r_j) o najniższym współczynniku SP . Jeśli więcej, niż dwie pary reguł mają ten sam SP , wybierz tą o mniejszych identyfikatorach.
3. Jeśli odrzucenie jakiegokolwiek z reguł spowodowałoby naruszenie poprawności zbioru, przejdź do punktu 2 i wybierz inną parę o najniższym współczynniku. Jeśli nie ma więcej par do wyboru, oznacza to, że odrzucenie jakiegokolwiek pary spowoduje naruszenie poprawności — zakończ działanie algorytmu. W przeciwnym przypadku przejdź do punktu 3.
4. Jeśli można odrzucić bez naruszenia poprawności tylko jedną z reguł, odrzuć ją. Jeśli można odrzucić obie — odrzuć tą regułę, która ma niższą wartość PP . Jeśli obie mają tę samą wartość PP , odrzuć tą o słabszej części decyzyjnej. W przypadku tych samych części decyzyjnych odrzuć tą, która ma niższy identyfikator
5. Uaktualnij zbiór R , usuwając z niego odrzuconą regułę.

Tablica 4.2: Algorytm selekcji reguł w oparciu o stopień przekrywania.

pełni” pokryły część obiektów (czyli obiekty te wliczają się do ich współczynników PP), to przy obliczaniu PP dla kolejnych reguł oczekujących na wejście do zbioru, nie bierzemy już pod uwagę tych obiektów. Wystarczy bowiem, że jedna z reguł pokryje w taki sposób przykład, aby spełnione były warunki poprawności. Działanie algorytmu kończy w momencie, gdy zbiór zacznie być poprawny.

4.3.5 Selekcja reguł na podstawie stopnia przekrywania

W celu wzbogacenia możliwości wyboru reguł definiujemy kolejną miarę, tym razem dotyczącą pary reguł, którą nazwiemy *stopniem przekrywania* (SP):

$$SP(r_i, r_j) = \frac{|[r_i] \cap [r_j]|}{|[r_i] \cup [r_j]|} \quad (4.24)$$

Oznacza on stopień zachodzenia reguł na siebie w sensie wspólnego wsparcia. Jest to stosunek liczby obiektów pokrywanych przez obie reguły do liczby obiektów pokrywanych przynajmniej przez jedną regułę. Tym samym świadczy on o tym, jak wiele przykładów obie reguły pokrywają wspólnie. Im wyższy SP , tym reguły bardziej podobne, gdyż mają wsparcie w postaci prawie tych samych obiektów.

Wybór takiej miary jest spowodowany faktem, że jeśli dwie reguły pokrywają bardzo podobne obiekty, to można przypuszczać, są do siebie bardzo podobne, a w związku z tym jedną z nich można odrzucić bez straty informacji o zbiorze danych. Z uwagi na fakt, że nie posługujemy się w sposób miarodajny odległościami między obiektami (wyłącznie związki monotoniczne), niestety nie jesteśmy w stanie tego podobieństwa oszacować w sposób bardziej ścisły i dokładny.

Na podstawie powyższej miary powstał algorytm (heurystyczny), znajdujący zbiór reguł jak najbardziej się od siebie różniących. Algorytm został przedstawiony w tablicy 4.2. Powinien został przeprowadzony osobno dla D^{\geq} -reguł, osobno dla D^{\leq} -reguł — nie porównuje się par reguł pochodzących z różnych typów, ponieważ ograniczają wartość wiarygodności decyzji w różnych kierunkach (z dołu lub z góry).

W przypadku podejści reguł gradualnych problemem jest niepowtarzalność klas (wiarygodności decyzji), co z kolei powoduje, że w przypadku gdy każdy obiekt cechuje się inną wartością wiarygodności decyzji, do zachowania warunków poprawności potrzebny jest zbiór tylu reguł, ile jest obiektów. Taki duży zbiór reguł utrudnia wyjaśnianie zbioru danych. Można zmniejszyć zbiór reguł poprzez zmianę warunku stopu na następujący: „przynajmniej $k\%$ obiektów musi zostać w pełni pokrytych ” (w pełni, tzn. przez reguły, przy których wartość progowa występująca w części decyzyjnej jest dokładnie równa dolnemu (górnemu) przybliżeniu takich obiektów). Oznacza to, że rezygnujemy z warunków poprawnego opisu, czyli opisujemy zbiór w sposób niedokładny, przybliżony. Jest to cena, jaką płaci się za zmniejszenie liczby reguł.

4.3.6 Przykład indukcji reguł

Indukcja reguł w oparciu o macierz dominacji zostanie przedstawiona na przykładzie z rozdziału 3.4 w celu pokazania uproszczenia opisu zbioru na skutek uogólnienia (indukcji). Nie będziemy tutaj rozważali redukcji atrybutów, ponieważ rozważany przykładowy zbiór danych ma tylko jeden trywialny redukt - zbiór wszystkich atrybutów. Poza tym, na skutek uogólniania poprzez redukcję reguły gradualne wyglądają dokładnie tak samo, za wyjątkiem faktu, że nie ma już w nich warunku na odrzuconych atrybutach. Ciekawsze wydają się wyniki działania algorytmów indukcji reguł.

Algorytm ADRIA, zastosowany z pewnymi modyfikacjami w celu rozwiązania przykładu, pochodzi z systemu *GOAL project* [11]. Po zastosowaniu algorytmu do wymienionego zbioru danych, utworzony został następujący wyczerpujący zbiór reguł minimalnych:

- D^{\geq} -reguły:

$$\begin{aligned}
r_1^{\geq} &: \mu_{X_1}(x) \geq 0.85 \rightarrow \mu_Y(x) \geq 0.75 \\
r_2^{\geq} &: \mu_{X_2}(x) \geq 0.8 \rightarrow \mu_Y(x) \geq 0.75 \\
r_3^{\geq} &: \mu_{X_1}(x) \geq 0.7 \wedge \mu_{X_2}(x) \geq 0.45 \rightarrow \mu_Y(x) \geq 0.6 \\
r_4^{\geq} &: \mu_{X_2}(x) \geq 0.6 \rightarrow \mu_Y(x) \geq 0.55 \\
r_5^{\geq} &: \mu_{X_1}(x) \geq 0.7 \rightarrow \mu_Y(x) \geq 0.5 \\
r_6^{\geq} &: \mu_{X_1}(x) \geq 0.55 \rightarrow \mu_Y(x) \geq 0.3 \\
r_7^{\geq} &: \mu_{X_2}(x) \geq 0.4 \rightarrow \mu_Y(x) \geq 0.3 \\
r_8^{\geq} &: \mu_{X_1}(x) \geq 0.35 \rightarrow \mu_Y(x) \geq 0.2 \\
r_9^{\geq} &: \mu_{X_2}(x) \geq 0.15 \rightarrow \mu_Y(x) \geq 0.2 \\
r_{10}^{\geq} &: \mu_{X_1}(x) \geq 0.2 \rightarrow \mu_Y(x) \geq 0.1 \\
r_{11}^{\geq} &: \mu_{X_2}(x) \geq 0.1 \rightarrow \mu_Y(x) \geq 0.1
\end{aligned}$$

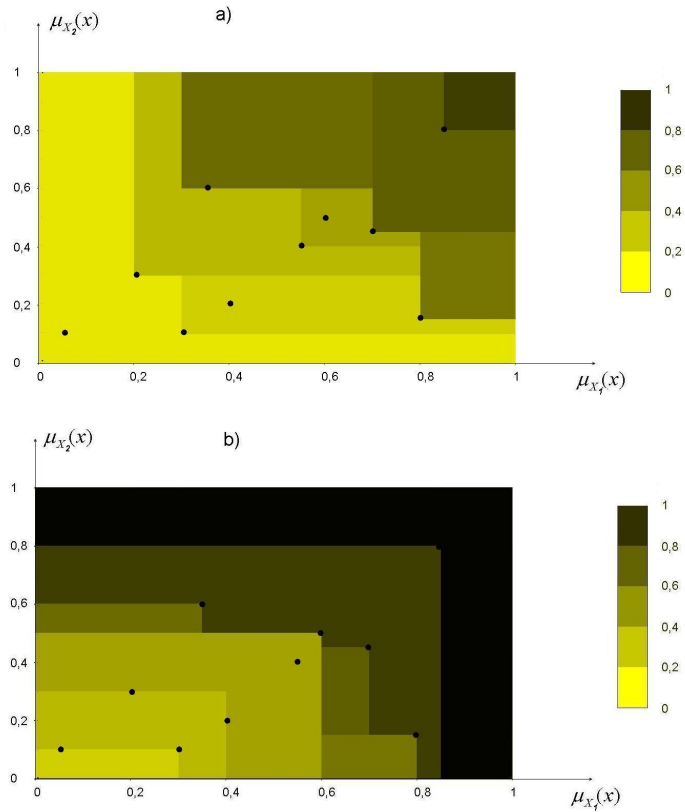
- D^{\leq} -reguły:

$$\begin{aligned}
r_1^{\leq} &: \mu_{X_1}(x) \leq 0.85 \rightarrow \mu_Y(x) \leq 0.75 \\
r_2^{\leq} &: \mu_{X_2}(x) \leq 0.8 \rightarrow \mu_Y(x) \leq 0.75 \\
r_3^{\leq} &: \mu_{X_2}(x) \leq 0.6 \rightarrow \mu_Y(x) \leq 0.7 \\
r_4^{\leq} &: \mu_{X_1}(x) \leq 0.8 \rightarrow \mu_Y(x) \leq 0.7 \\
r_5^{\leq} &: \mu_{X_1}(x) \leq 0.4 \rightarrow \mu_Y(x) \leq 0.65 \\
r_6^{\leq} &: \mu_{X_2}(x) \leq 0.3 \rightarrow \mu_Y(x) \leq 0.5 \\
r_7^{\leq} &: \mu_{X_1}(x) \leq 0.2 \rightarrow \mu_Y(x) \leq 0.25 \\
r_8^{\leq} &: \mu_{X_1}(x) \leq 0.4 \wedge \mu_{X_2}(x) \leq 0.3 \rightarrow \mu_Y(x) \leq 0.25 \\
r_9^{\leq} &: \mu_{X_2}(x) \leq 0.1 \rightarrow \mu_Y(x) \leq 0.1
\end{aligned}$$

Mamy więc jedenaście D^{\geq} -reguł i dziewięć D^{\leq} -reguł - łącznie dwadzieścia. Jest ich dwukrotnie więcej niż obiektów. Jednak należy pamiętać, że zbiór jest wyczerpujący i prawdopodobnie nadmiarowy. Wykorzystamy teraz dwa wymienione poprzednio podejścia do oceny reguł, aby spróbować odrzucić jakies z nich.

Zacznijmy od podejścia pierwszego. W tabeli 4.3 umieszczono wartości współczynników względnego i pełnego wsparcia, obliczone dla każdej z reguł. Wybierać będziemy reguły według współczynnika PP , a dopiero, jeśli dwie reguły mają tę samą wartość, użyjemy WP . W przypadku równych WP , o wyborze decyduje niższy identyfikator. Po każdym kroku należy na nowo przeliczyć współczynniki PP , ignorując obiekty, które zostały wliczone do PP wybranych już reguł. W ten sposób spośród D^{\geq} -reguł wybierzemy następujące (w tej kolejności): r_9^{\geq} , r_7^{\geq} , r_4^{\geq} , r_1^{\geq} , r_3^{\geq} , r_{10}^{\geq} , r_5^{\geq} , natomiast wśród D^{\leq} -reguł będą to: r_3^{\leq} , r_8^{\leq} , r_5^{\leq} , r_1^{\leq} , r_6^{\leq} , r_9^{\leq} .

W wyniku selekcji udało się wybrać trzynaście reguł, które poprawnie opisują zbiór. Liczby tej nie można zmniejszyć z uwagi na fakt, że tyle jest właśnie różnych wartości wiarygodności górnego i dolnego przybliżenia, stąd zmniejszenie liczby reguł musiałoby spowodować, że jakiś przykład nie zostanie pokryty



Rysunek 4.1: Graficzne przedstawienie modelu regułowego (zaciemnienie obszaru odpowiada wartościom wiarygodności decyzji) dla D^{\geq} -reguł (a) oraz D^{\leq} -reguł (b).

„w pełni”, czyli przez regułę o parametrze β części decyzyjnej tak silnym jak jego wiarygodność dolnego (górnego) przybliżenia.

Następnie przejdziemy do przetestowania drugiego algorytmu, opartego na stopniu pokrywania. Rozpoczynamy od pełnego zbioru i wybieramy reguły z największym stopniem pokrywania, po czym jedną z nich odrzucamy (tą, która ma najmniejszy współczynnik PP , w przypadku równych współczynników — słabszą część decyzyjną, w przypadku równych części decyzyjnych — o wyższym identyfikatorze). W ten sposób spośród D^{\geq} -reguł odrzucimy następujące (w tej kolejności): r_1^{\geq} (niższy identyfikator), r_{10}^{\geq} , r_8^{\geq} , r_7^{\geq} , natomiast wśród D^{\leq} -reguł odrzucimy: r_1^{\leq} (najwyższy identyfikator), r_3^{\leq} , r_8^{\leq} (gdyż żadna z pary reguł (r_2^{\leq} , r_4^{\leq}) nie mogła zostać odrzucona z powodu utraty poprawności).

Tym samym w zbiorze zostaną:

Reguła	WP	PP
r_1^{\succ}	1	1
r_2^{\succ}	1	1
r_3^{\succ}	1	1
r_4^{\succ}	0.75	2
r_5^{\succ}	0.6	1
r_6^{\succ}	0.71	2
r_7^{\succ}	0.86	2
r_8^{\succ}	0.78	1
r_9^{\succ}	1	2
r_{10}^{\succ}	1	1
r_{11}^{\succ}	1	1
r_1^{\preceq}	1	1
r_2^{\preceq}	1	1
r_3^{\preceq}	1	3
r_4^{\preceq}	1	3
r_5^{\preceq}	0.83	2
r_6^{\preceq}	1	1
r_7^{\preceq}	0.33	1
r_8^{\preceq}	1	2
r_9^{\preceq}	1	1

Tablica 4.3: Ocena reguł za pomocą względnego wsparcia (WP) i pełnego wsparcia (PP)

- D^{\succ} -reguły: $r_2^{\succ}, r_3^{\succ}, r_4^{\succ}, r_5^{\succ}, r_6^{\succ}, r_9^{\succ}, r_{11}^{\succ}$;
- D^{\preceq} -reguły: $r_2^{\preceq}, r_4^{\preceq}, r_5^{\preceq}, r_6^{\preceq}, r_7^{\preceq}, r_9^{\preceq}$.

Zbiór ten jest równoliczny ze zbiorem wybranym przez poprzedni algorytm, gdyż jest to minimalny możliwy zbiór reguł. Zarazem zbiory utworzone przez oba algorytmy są do siebie w tym przypadku bardzo podobne. Co więcej, w tym wypadku trudno stwierdzić jednoznacznie, który ze zbiorów jest lepszy. Oba są minimalnymi zbiorami reguł minimalnych, ale na skutek dużej ilości klas (wartości wiarygodności decyzji) istnieje wiele zbiorów o takiej własności. Jakąkolwiek podstawą do wysunięcia wniosków o przewadze jednego algorytmu nad drugim byłby dopiero eksperyment obliczeniowy wyznaczający jakość klasyfikacji na zbiorze testującym dla wielu różnych instancji.

Na koniec podsumujemy rozważania. W podrozdziale 3.4, przy okazji analizowania przykładowego zbioru, pokazano, że można reprezentować reguły (funkcje) gradualne w postaci zbioru implikacji (reguł). W poprzednich podrozdziałach staraliśmy się pokazać, że zbiór indukowanych reguł decyzyjnych opisuje dane poprawnie, stąd zgodnie z definicją z 3.2.3 *jest równoważny* funkcjom gra-

	r_1^{\geq}	r_2^{\geq}	r_3^{\geq}	r_4^{\geq}	r_5^{\geq}	r_6^{\geq}	r_7^{\geq}	r_8^{\geq}	r_9^{\geq}	r_{10}^{\geq}	r_{11}^{\geq}
r_1^{\geq}	-	1	0.5	0.33	0.33	0.2	0.17	0.14	0.11	0.1	0.1
r_2^{\geq}	1	-	0.5	0.33	0.33	0.2	0.17	0.14	0.11	0.1	0.1
r_3^{\geq}	0.5	0.5	-	0.25	0.67	0.4	0.33	0.29	0.22	0.2	0.2
r_4^{\geq}	0.33	0.33	0.25	-	0.2	0.14	0.5	0.25	0.33	0.3	0.3
r_5^{\geq}	0.33	0.33	0.67	0.2	-	0.6	0.29	0.43	0.33	0.3	0.3
r_6^{\geq}	0.2	0.2	0.4	0.14	0.6	-	0.57	0.71	0.56	0.5	0.5
r_7^{\geq}	0.17	0.17	0.33	0.5	0.29	0.57	-	0.62	0.67	0.6	0.6
r_8^{\geq}	0.14	0.14	0.29	0.25	0.43	0.71	0.62	-	0.78	0.7	0.7
r_9^{\geq}	0.11	0.11	0.22	0.33	0.33	0.56	0.67	0.78	-	0.9	0.9
r_{10}^{\geq}	0.1	0.1	0.2	0.3	0.3	0.5	0.6	0.7	0.9	-	1
r_{11}^{\geq}	0.1	0.1	0.2	0.3	0.3	0.5	0.6	0.7	0.9	1	-

Tablica 4.4: Stopnie przekrywania dla D^{\geq} -regul

	r_1^{\leq}	r_2^{\leq}	r_3^{\leq}	r_4^{\leq}	r_5^{\leq}	r_6^{\leq}	r_7^{\leq}	r_8^{\leq}	r_9^{\leq}
r_1^{\leq}	-	1	0.9	0.9	0.5	0.4	0.1	0.3	0.1
r_2^{\leq}	1	-	0.9	0.9	0.5	0.4	0.1	0.3	0.1
r_3^{\leq}	0.9	0.9	-	1	0.56	0.44	0.11	0.33	0.11
r_4^{\leq}	0.9	0.9	1	-	0.56	0.44	0.11	0.33	0.11
r_5^{\leq}	0.5	0.5	0.56	0.56	-	0.5	0.2	0.6	0.2
r_6^{\leq}	0.4	0.4	0.44	0.44	0.5	-	0.25	0.75	0.25
r_7^{\leq}	0.1	0.1	0.11	0.11	0.2	0.25	-	0.33	0
r_8^{\leq}	0.3	0.3	0.33	0.33	0.6	0.75	0.33	-	0.33
r_9^{\leq}	0.1	0.1	0.11	0.11	0.2	0.25	0	0.33	-

Tablica 4.5: Stopnie przekrywania dla D^{\leq} -regul

dualnym dolnego i górnego przybliżenia. Wszystkie wymienione dotychczas sposoby ogólniania zbioru danych prowadzą do funkcji (regul) gradualnych, zdefiniowanych w rozdziale 3 z pominięciem warunku maksymalnej ostrożności.

Można jednak posunąć się jeszcze dalej i zrezygnować również z dokładnego spełniania warunku poprawności przyjmując, że utworzony opis zbioru będzie z tym zbiorem spójny jedynie w przybliżeniu. Taki opis, zwykle prostszy (nakładający mniejsze ograniczenia), może być satysfakcjonujący w rzeczywistych problemach, tym bardziej, że zwykle dane obarczone są błędami. Jednak rozważania przy użyciu terminów takich, jak „w przybliżeniu” czy „niewielki błąd” sugerują zastosowanie do budowy opisu skali silniejszej, niż tylko porządkowa, wymagającej posługiwania się bezpośrednio odległościami. Skala taka nazywana jest *ilorazową*.

Rozdział 5

Opis zbioru danych w oparciu o skalę ilorazową

5.1 Uwagi wstępne

W poprzednich rozdziałach rozważano metody indukcji reguł gradualnych wyłącznie przy wykorzystaniu własności porządkowych (monotoniczności) danych na wszystkich kryteriach, nie uwzględniając wartości numerycznych, co z kolei pozwoliłoby porównywać wzajemne odległości między obiektami. Wybór odpowiedniej metody postępowania jest tak naprawdę równoważny z wyborem skali preferencji (rodzaje skal wymienione są np. w [19]) — przyjmując, że odległości między obiektami nie mają znaczenia, stosujemy skalę *porządkową*. Z kolei porównywanie odległości prowadzi do skali *ilorazowej*. Zgodnie z *zasadą maksymalnej ostrożności* [15] staraliśmy się przyjąć, że wartości reguł (funkcji) gradualnych w punktach w przestrzeni kryteriów, gdzie leżą obiekty, są równe wartościom dolnych (lub górnych) przybliżeń tych obiektów, natomiast gdy w danym punkcie żadnego obiektu nie ma, wnioskujemy o wartości funkcji wyłącznie na podstawie wiedzy o monotoniczności, przyjmując wartość jako równą największemu z dolnych (górnym) przybliżeń, szukanemu wśród tych obiektów, które taki punkt dominują. Takie postępowanie odpowiada opisaniu zbioru danych zgodnie ze skalą porządkową.

Zauważmy jednak, że w podejściu opartym na regułach gradualnych, wszystkie kryteria i decyzja są wyrażone w postaci liczb rzeczywistych z przedziału $[0, 1]$, będących stopniami przynależności do pewnych zbiorów rozmytych. Wartości te otrzymywane są po pewnych przekształceniach oryginalnego zbioru danych, ale mimo tego, że są liczbami rzeczywistymi, nadal traktujemy je jako zbiór symboli, które potrafimy wzajemnie uszeregować, ale nie możemy wypowiadać się o odległościach między nimi. Skoro mamy jednak liczby, nasuwa się pomysł, aby fakt ten wykorzystać i wprowadzić skalę ilorazową. Zaletą będzie tutaj możliwość wykorzystania miar bazujących na wzajemnej bliskości dwóch obiektów lub różnicy w ich wartościach wiarygodności decyzji. Można również dokonać

szacowania wartości w punktach, o których nie mamy bezpośrednich informacji z tablicy decyzyjnej, poprzez ekstrapolację pewnej funkcji trendu (regresję). Wadą z kolei jest przyjęcie znacznie silniejszych założeń, być może koncepcyjnie zbyt silnych przy rozpatrywaniu problemów rzeczywistych [14].

W tym rozdziale rozważane są dwie metody uogólniania opisu zbioru danych. Pierwsza opiera się na indukcji reguł opisanej w rozdziale poprzednim i dotyczy wprowadzenia nowych, silniejszych miar jakości reguł w celu dokonania selekcji, dużo mocniejszej niż w poprzednio. Drugi dotyczy aproksymacji zbioru danych za pomocą funkcji odcinkami liniowej.

5.2 Minimalizacja zbioru reguł w oparciu o odległości między obiektami

Jednym z podstawowych problemów pojawiających się w przypadku podejścia opartego na regułach gradualnych, jest duża (teoretycznie nieprzeliczalna) liczba możliwych wartości, jakie może przyjąć wiarygodność decyzji obiektów. Ponieważ koncepcyjnie wiarygodność decyzji odpowiada klasom, w efekcie mamy do czynienia z problemem bardzo drobnej klasyfikacji (wiele klas zawierających nieliczne obiekty). Pociąga to za sobą skomplikowany opis takiego zbioru za pomocą reguł lub w jakikolwiek inny sposób, o ile traktujemy wiarygodności decyzji jako symbole podlegające relacji porządku. Gdybyśmy jednak potraktowali je jako liczby o określonych wartościach, otwiera się szereg metod pozwalających zwiększyć ziarnistość klasyfikacji. Dokonać tego można poprzez dyskretyzację wartości (podział ciągłej przestrzeni decyzji lub kryteriów na odcinki o mierzalnej długości, co prowadzi do pojawienia się zbiorów przeliczalnych), analizę skupień (na przykład, zastępowanie dużych skupisk obiektów za pomocą jednego obiektu odzwierciedlającego ich średnie cechy) lub dowolny inny sposób pozwalający uprościć opis zbioru.

W przypadku reguł będzie to mocniejsza selekcja. Problemem pojawiającym się przy indukcji jest fakt, że ich liczba rośnie proporcjonalnie do liczby klas, a tym samym do liczby obiektów, co z kolei powoduje, że opis zbioru za pomocą reguł jest obszerniejszy niż opis za pomocą obiektów, trudny do interpretacji. W rzeczywistości bardzo często sprawdza się reguła mówiąca, że opis bardziej prosty zwykle jest bardziej wiarygodny i poprawny (zasada tzw. brzytwy Ockhama [3]). Aby zmniejszyć złożoność tkwiącą w dużym zbiorze reguł, proponujemy w tym rozdziale podejście, pozwalające ignorować małe różnice między obiektami, zarówno na skali kryteriów, jak i decyzji.

5.2.1 Miara jakości reguł na podstawie odległości

Rozważmy dwa obiekty, leżące blisko siebie w przestrzeni kryteriów, co oznacza, że wiarygodności ich przynależności (które traktujemy jako kryteria) do zbiorów rozmytych opartych na wszystkich atrybutach warunkowych niewiele się różnią. Obiekty mogą się natomiast nawet znacznie różnić wiarygodnością decyzji. Przyjmijmy, że pewien algorytm indukcji reguł postawi reguły oparte

na tych obiektach. Tym samym części warunkowe tych reguł również niewiele się od siebie różnią. Nasuwa się pytanie — jaki sens ma stawianie dwóch reguł tak blisko siebie, skoro odległość między nimi jest na tyle mała, że niemal zaniebdywalna i w rzeczywistych problemach nie powinna mieć wpływu na decyzję dotyczącą tych przypadków. Trudno przypuszczać (być może poza skomplikowanymi i nieliniowymi systemami), że minimalna zmiana w przestrzeni kryteriów miałaby spowodować znaczącą zmianę decyzji. Tym bardziej powinno tak w być w przypadku wspomagania decyzji — decydent reaguje na znaczne, widoczne zmiany na kryteriach, drobne w naturalny sposób zaniebdując. W związku z tym przyjęto, że dwa obiekty, leżące blisko siebie w przestrzeni kryteriów, są do siebie bardzo podobne.

Inna sytuacja ma miejsce w przypadku, gdy dwa obiekty, być może oddalone od siebie, mają niewiele różniące się wartości wiarygodności decyzji. Z uwagi na fakt, że małe różnice w rzeczywistości są zazwyczaj zaniebdywalne (szczególnie w przypadku wspomagania decyzji), można wartość decyzji dla takich obiektów ze sobą utożsamić.

Zauważmy, że pierwsza z opisanych sytuacji odpowiada w pewnym sensie przypadkowi dyskretyzacji dokonanej w przestrzeni kryteriów (podobne wartości na kryterium należą zazwyczaj do tego samego przedziału, więc stają się takie same), natomiast druga z nich — dyskretyzacji na skali wiarygodności decyzji.

Z uwagi na to, że przestrzeń kryteriów jest z reguły wielowymiarowa, rozpozniemy od wprowadzenia metryki w niej obowiązującej. Przyjmijmy, że mamy dwa obiekty, które można utożsamić z punktami w przestrzeni wielowymiarowej: $\mathbf{x} = \{x_1, \dots, x_n\}$ oraz $\mathbf{y} = \{y_1, \dots, y_n\}$, gdzie n to ilość kryteriów ($n = r + s$ dla rodziny zbiorów $\mathcal{X} = \{X_{i_1}^\uparrow, \dots, X_{i_r}^\uparrow, X_{j_1}^\downarrow, \dots, X_{j_s}^\downarrow\}$). Wyróżnia się kilka podstawowych metryk:

- „miejska” (Lp_1):

$$d(x, y) = \sum_{i=1}^n |x_i - y_i| \quad (5.1)$$

- euklidesowa (Lp_2):

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (5.2)$$

- Czebyszewa (Lp_∞):

$$d(x, y) = \lim_{p \rightarrow \infty} \sqrt[p]{\sum_{i=1}^n (x_i - y_i)^p} = \max_i |x_i - y_i| \quad (5.3)$$

Przyjmijemy w tym rozdziale metrykę Czebyszewa zakładając, że znaczenie ma największa z różnic na kryteriach występujących między dwoma obiektami. Mimo to dobór metryki jest w dużej mierze arbitralny.

Rozważmy D^{\geq} -regulę gradualną postaci:

$$\begin{aligned} \text{jeśli } \mu_{X_{i_1}^\uparrow}(x) \geq \alpha_{i_1} \wedge \cdots \wedge \mu_{X_{i_m}^\uparrow}(x) \geq \alpha_{i_m} \wedge \mu_{X_{j_1}^\downarrow}(x) \leq \alpha_{j_1} \wedge \cdots \wedge \\ \wedge \mu_{X_{j_m}^\downarrow}(x) \leq \alpha_{j_m} \text{ to } \mu_Y(x) \geq \beta \end{aligned} \quad (5.4)$$

Może to być również D^{\leq} -reguła gradualna:

$$\begin{aligned} \text{jeśli } \mu_{X_{i_1}^\uparrow}(x) \leq \alpha_{i_1} \wedge \cdots \wedge \mu_{X_{i_m}^\uparrow}(x) \leq \alpha_{i_m} \wedge \mu_{X_{j_1}^\downarrow}(x) \geq \alpha_{j_1} \wedge \cdots \wedge \\ \wedge \mu_{X_{j_m}^\downarrow}(x) \geq \alpha_{j_m} \text{ to } \mu_Y(x) \leq \beta \end{aligned} \quad (5.5)$$

Regulę obu typów można utożsamić z parą (α, β) , gdzie $\alpha = [\alpha_{i_1}, \dots, \alpha_{i_m}, \alpha_{j_1}, \dots, \alpha_{j_m}]$. Wprowadzimy teraz zbiór, który nazwiemy *otoczeniem wspierającym*, definiując je dla D^{\geq} -reguły (α, β) jako zbiór obiektów $x \in U$, które są pokryte przez regulę (czyli spełniają jej część warunkową) oraz spełniających co najmniej jeden z następujących warunków:

$$d(x, \alpha) \leq \gamma \quad (5.6)$$

$$0 \leq \mu[\underline{App}(\mathcal{X}, Y), x] - \beta \leq \lambda \quad (5.7)$$

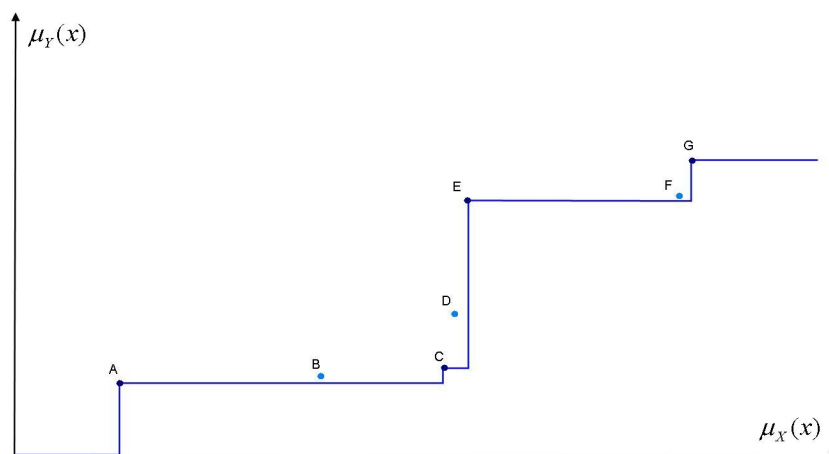
Dla D^{\leq} -reguł warunek (5.6) pozostanie bez zmian, natomiast warunek (5.7) przyjmie postać:

$$0 \leq \beta - \mu[\overline{App}(\mathcal{X}, Y), x] \leq \lambda \quad (5.8)$$

Gdzie λ i γ to pewne progi, przyjęte jako parametry algorytmu. W skład otoczenia wspierającego reguły (α, β) wchodzi więc obiekty, które leżą bardzo blisko jej punktu zaczepienia (α) lub ich decyzja nieznacznie różni się od parametry z części decyzyjnej reguły (β) . Warunek pierwszy odpowiada koncepcyjnie dyskretyzacji w przestrzeni kryteriów, warunek drugi — w przestrzeni decyzji. Podstawowe znaczenie będzie miała ?? dla reguły r , którą będziemy oznaczać $N(r)$. W przypadku przyjęcie progów $\gamma = \lambda = 0$ powyższa definicja liczności otoczenia będzie równoważna z definicją pełnego wsparcia (PW) z 4.3.4. Interpretacja powyższej miary została przedstawion na rysunku 5.1.

5.2.2 Selekcja reguł na podstawie odległości

Mając zdefiniowaną miarę liczności otoczenia wspierającego możemy na jej podstawie dokonać selekcji reguł. Przyjmijmy, że mamy wyczerpujący zbiór



Rysunek 5.1: Wykres wiarygodności decyzji w funkcji wiarygodności kryterium, powstały na skutek zastosowania reguł. W punktach, gdzie następuje skok, tzn. wykres wznosi się pionowo w górę (punkty A, C, E, G), zostały umiejscowione reguły. Nie ma sensu umieszczanie reguły w punkcie B, gdyż mimo że jest oddalony od A na osi kryterium, jest bardzo blisko tego punktu na osi decyzji (podobnie dla punktu F). Z kolei punkt D jest blisko punktu C na osi kryterium, mimo stosunkowo dużej odległości na osi decyzji, stąd tam również nie stawiamy reguły.

reguł minimalnych. Algorytm, przedstawiony w tabeli 5.1, jest zachłanną heurystyką i polega na wyborze takich reguł, które mają największe $N(r)$. Następnie za każdym razem uaktualniamy $N(r)$ wszystkich dotąd niewybranych reguł, nie wliczając już do niego przykładów, które weszły w skład wsparcia reguł wybranych (skoro staramy się użyć jak najmniejszej ilości reguł, nie ma sensu pokrywać danego przykładu wieloma regułami). Powtarzając tę czynność do czasu, aż wszystkie obiekty zostaną pokryte (zgodnie z powyższą definicją stosowanej miary), otrzymujemy zbiór reguł wyjaśniający dane, o rozmiarze uzależnionym od wyboru progów γ i λ . Reguły takie nie pokrywają wszystkich obiektów w najsilniejszy możliwy sposób, ale z pewnym przybliżeniem związanym z zaniedbywaniem małych odległości na skali.

Tym samym, poprzez zastosowanie wartości progowych, otrzymujemy rzadszy podział na klasy z uwagi na fakt, że bliskie sobie wartości wiarygodności decyzji utożsamiane są teraz ze sobą. Prowadzi to z kolei do prostszego i ogólniejszego opisu. W przypadku stosunkowo dużych różnic w wiarygodności decyzji i względnie równomiernego rozstawienia obiektów (co powoduje, że nie leżą z reguły blisko siebie w przestrzeni kryteriów), opisana metoda daje gorsze rezultaty, ale w rzeczywistych problemach nie zdarza się to za często. Mimo

Wejście: Wyczerpujący zbiór reguł R

Wyjście: Zredukowany zbiór reguł R' w przybliżeniu pokrywający wszystkie przykłady

Algorytm selekcji reguł w oparciu o porównywanie odległości między obiektami:

1. Przyjmij $R' = \emptyset$ oraz $U' = U$ (tymczasowy zbiór niepokrytych obiektów)
2. Wyznacz dla każdej reguły r ze zbioru R miarę liczności otoczenia wspierającego $N(r)$ na podstawie U' .
3. Ze zbioru R wybierz regułę r_l o największym współczynniku $N(r_l)$ i przenieś ją do zbioru R' , usuwając z R : $R = R - \{r_l\}$, $R' = R' \cup \{r_l\}$
4. Usuń ze zbioru niepokrytych obiektów U' zbiór obiektów U_l , które zostały wliczone do miary $N(r_l)$ reguły r_l (czyli wspierających tą regułę): $U' = U' - U_l$.
5. jeśli $U' = \emptyset$, zakończ działanie algorytmu. W przeciwnym przypadku wróć do punktu 2.

Tablica 5.1: Algorytm selekcji reguł w oparciu o porównywanie odległości między obiektami

to nadal opis zbioru pozostaje opisem regułowym. Będąc w przestrzeni numerycznej, można się doszukiwać pewnych zależności między kryteriami a decyzją, które lepiej wyrazić za pomocą ciągłej, monotonicznej funkcji (w odróżnieniu od reguł, gdyż funkcja z nich złożona posiada nieciągłości we wszystkich punktach zaczepienia reguł). Problem taki zostanie rozpatrzony w dalszej części rozdziału.

5.2.3 Przykład selekcji reguł

Rozważmy ten sam przykład, jaki został przedstawiony w poprzednich rozdziałach. Ponownie wypiszemy reguły, jakie zostałyby wyindukowane z użyciem dowolnego algorytmu, generującego wyczerpujący zbiór reguł oparty na przykładach:

- D^{\geq} -reguły:

$$r_1^{\geq}: \quad \mu_{X_1}(x) \geq 0.85 \rightarrow \mu_Y(x) \geq 0.75$$

$$\begin{aligned}
r_2^{\geq} &: \mu_{X_2}(x) \geq 0.8 \rightarrow \mu_Y(x) \geq 0.75 \\
r_3^{\geq} &: \mu_{X_1}(x) \geq 0.7 \wedge \mu_{X_2}(x) \geq 0.45 \rightarrow \mu_Y(x) \geq 0.6 \\
r_4^{\geq} &: \mu_{X_2}(x) \geq 0.6 \rightarrow \mu_Y(x) \geq 0.55 \\
r_5^{\geq} &: \mu_{X_1}(x) \geq 0.7 \rightarrow \mu_Y(x) \geq 0.5 \\
r_6^{\geq} &: \mu_{X_1}(x) \geq 0.55 \rightarrow \mu_Y(x) \geq 0.3 \\
r_7^{\geq} &: \mu_{X_2}(x) \geq 0.4 \rightarrow \mu_Y(x) \geq 0.3 \\
r_8^{\geq} &: \mu_{X_1}(x) \geq 0.35 \rightarrow \mu_Y(x) \geq 0.2 \\
r_9^{\geq} &: \mu_{X_2}(x) \geq 0.15 \rightarrow \mu_Y(x) \geq 0.2 \\
r_{10}^{\geq} &: \mu_{X_1}(x) \geq 0.2 \rightarrow \mu_Y(x) \geq 0.1 \\
r_{11}^{\geq} &: \mu_{X_2}(x) \geq 0.1 \rightarrow \mu_Y(x) \geq 0.1
\end{aligned}$$

• D^{\leq} -reguły:

$$\begin{aligned}
r_1^{\leq} &: \mu_{X_1}(x) \leq 0.85 \rightarrow \mu_Y(x) \leq 0.75 \\
r_2^{\leq} &: \mu_{X_2}(x) \leq 0.8 \rightarrow \mu_Y(x) \leq 0.75 \\
r_3^{\leq} &: \mu_{X_2}(x) \leq 0.6 \rightarrow \mu_Y(x) \leq 0.7 \\
r_4^{\leq} &: \mu_{X_1}(x) \leq 0.8 \rightarrow \mu_Y(x) \leq 0.7 \\
r_5^{\leq} &: \mu_{X_1}(x) \leq 0.4 \rightarrow \mu_Y(x) \leq 0.65 \\
r_6^{\leq} &: \mu_{X_2}(x) \leq 0.3 \rightarrow \mu_Y(x) \leq 0.5 \\
r_7^{\leq} &: \mu_{X_1}(x) \leq 0.2 \rightarrow \mu_Y(x) \leq 0.25 \\
r_8^{\leq} &: \mu_{X_1}(x) \leq 0.4 \wedge \mu_{X_2}(x) \leq 0.3 \rightarrow \mu_Y(x) \leq 0.25 \\
r_9^{\leq} &: \mu_{X_2}(x) \leq 0.1 \rightarrow \mu_Y(x) \leq 0.1
\end{aligned}$$

Przyjmijmy, że wartości progów są następujące: $\gamma = 0.1$, $\lambda = 0.05$. Do mierzenia odległości użyjemy metrykę Czebyszewa. Wsparcie poszczególnych reguł zostało przedstawione w tabeli 5.2

D^{\geq} -reguła	liczność otoczenia	D^{\leq} -reguła	liczność otoczenia
r_1^{\geq}	1	r_1^{\leq}	5
r_2^{\geq}	1	r_2^{\leq}	4
r_3^{\geq}	1	r_3^{\leq}	5
r_4^{\geq}	2	r_4^{\leq}	6
r_5^{\geq}	2	r_5^{\leq}	4
r_6^{\geq}	2	r_6^{\leq}	3
r_7^{\geq}	3	r_7^{\leq}	1
r_8^{\geq}	2	r_8^{\leq}	2
r_9^{\geq}	3	r_9^{\leq}	1
r_{10}^{\geq}	3		
r_{11}^{\geq}	2		

Tablica 5.2: Ocena reguł za pomocą wsparcia

Sposób D^{\geq} -reguł wybierzemy następujące (w tej kolejności): $r_7^{\geq}, r_9^{\geq}, r_4^{\geq}, r_1^{\geq}, r_{10}^{\geq}$, natomiast wśród D^{\leq} -reguł będą to: $r_4^{\leq}, r_5^{\leq}, r_1^{\leq}, r_6^{\leq}, r_9^{\leq}$. Zauważmy, że liczba reguł jest mniejsza niż w przypadku selekcji z wykorzystaniem wyłącznie własności porządkowych. Generalnie, im większe progi δ i λ , tym liczba reguł spada.

5.3 Problem regresji wielowymiarowej za pomocą funkcji odcinkami liniowej

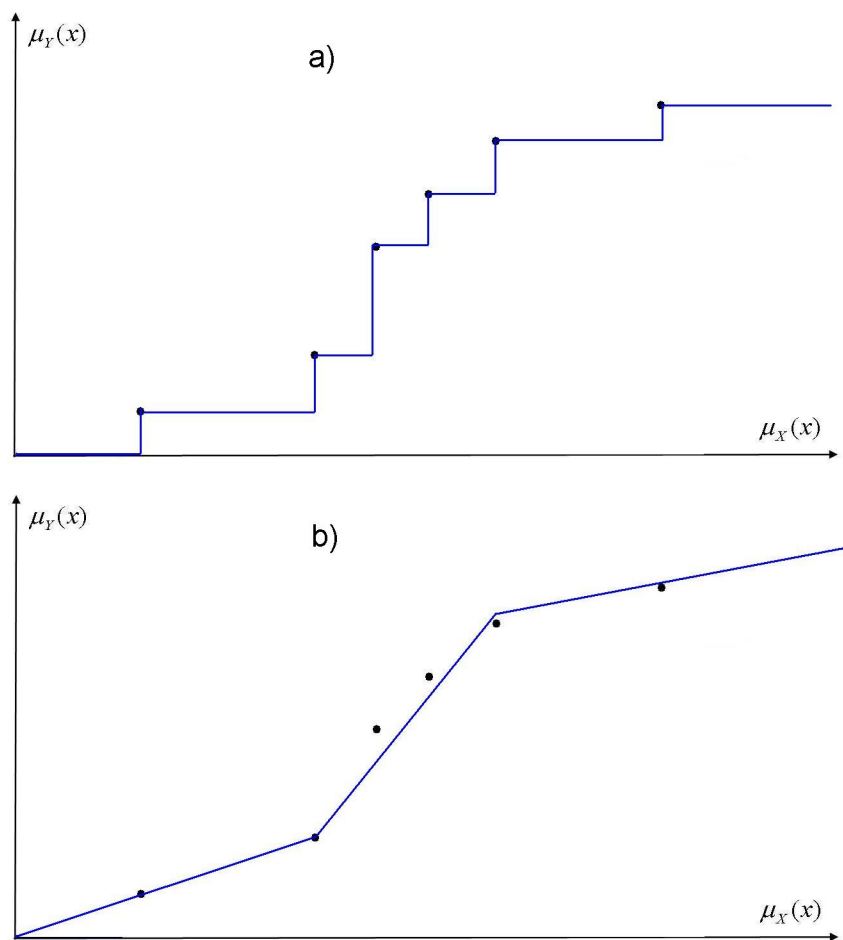
Jeśli zdecydowalibyśmy się na aproksymację zbioru za pomocą funkcji, dobierając odpowiedni jej kształt (np. liniowy) możemy starać się metodą regresji dopasować funkcję do punktów o znanych wartościach decyzji z najmniejszym możliwym błędem, dzięki czemu otrzymujemy informacje o zbiorze danych wyrażone w postaci wzoru takiej funkcji. Pozwala to wyznaczyć interesującą nas wartość decyzji w każdym punkcie przestrzeni. Na rysunku 5.2 pokazano dwa sposoby wyznaczania wartości funkcji - za pomocą zasady maksymalnej ostrożności (zwykłych reguł gradualnych) oraz aproksymacji funkcją odcinkami liniową. W drugim przypadku wartości szacowane są mniej ostrożnie, wykorzystuje się jednak fakt liniowego wzrostu wartości na pewnych odcinkach, czyli dopasowania do pewnego modelu (w tym przypadku: liniowego). Jest to również rodzaj indukcji, uogólnienie następuje jednak poprzez przypisanie danym zależności funkcyjnej.

Zostanie więc przeprowadzona na zbiorze regresja, za pomocą funkcji odcinkami liniowej, a ściślej mówiąc — funkcji będącej sumą funkcji cząstkowych (zależnych tylko od wartości na jednym kryterium), przy czym każda z nich ma postać odcinkami liniową. Pokazany zostanie sposób tworzenia dwóch funkcji - dolnego i górnego przybliżenia, odpowiednio $f(\cdot)$ i $g(\cdot)$. Pierwsza z nich przyjmuje dla obiektów z tablicy decyzyjnej wartości nie większe niż wartość wiarygodności dolnego przybliżenia decyzji tych obiektów, stąd szacuje zbiór od dołu. Druga z nich przyjmuje wartości nie mniejsze niż wiarygodność górnego przybliżenia decyzji, stąd jest oszacowaniem od góry. Zbiór obiektów powinien zawierać się w pasie przestrzeni ograniczonym obiema funkcjami. Oczywiście, aby zachować zgodność z zasadą dominacji, funkcje muszą być rosnące na kryteriach typu *zysk* oraz malejące na kryteriach typu *koszt*.

5.3.1 Programowanie matematyczne

Zanim przejdziemy do dalszej analizy, należy jeszcze przypomnieć pewną ogólną metodę rozwiązywania problemów optymalizacyjnych, znaną jako *programowanie matematyczne*. Polega ona na minimalizacji (maksymalizacji) pewnej funkcji przy ograniczeniach:

$$\begin{aligned} \min z &= f(\mathbf{x}) \\ \text{przy ogr.: } &\mathbf{g}(\mathbf{x}) \leq \mathbf{0} \quad f: \mathbb{R}^n \rightarrow \mathbb{R}, \quad \mathbf{g}: \mathbb{R}^n \rightarrow \mathbb{R}^m \end{aligned}$$



Rysunek 5.2: Aproxymacja punktów (problem jednokryterialny) za pomocą reguł gradualnych(a) oraz funkcji odcinkami liniowej (b).

Szczególny przypadek, w którym funkcje $f(\cdot)$ i $g(\cdot)$ są liniowe, nazywamy programowaniem liniowym. Najpopularniejszą metodą rozwiązywania tego typu zagadnienia jest *metoda sympleks*, po raz pierwszy podana przez G. Danziga [4]. Polega na przeszukiwaniu punktów wierzchołkowych obszaru dopuszczalnego (spełniającego ograniczenia) w przestrzeni zmiennych decyzyjnych, przechodząc do wierzchołka sąsiedniego z równoczesnym polepszeniem funkcji celu. Można pokazać [1], że rozwiązanie optymalne znajduje się w jednym z punktów wierzchołkowych (funkcja nie ma ekstremum w zbiorze). Przejście między wierzchołkami odbywa się poprzez zmianę bazy — zastąpienie jednego z

wektorów bazowych innym w ten sposób, że funkcja celu ulega poprawie. Ostatecznie algorytm zatrzymuje się w wierzchołku, z którego nie ma możliwości wykonania kroku niosącego poprawę — jest to rozwiązanie optymalne.

Metoda sympleks ma złożoność wykładniczą, ponieważ liczba wierzchołków wielościanu rozwiązań dopuszczalnych rośnie wykładniczo, a w najbardziej pesymistycznym przypadku należy odwiedzić wszystkie z nich. W sytuacjach rzeczywistych działa jednak bardzo dobrze i szybko zbiega do rozwiązania optymalnego. Mimo, że sam problem programowania liniowego należy do klasy P (problemów rozwiązywanych w liczbie kroków będącej wielomianem od rozmiaru instancji [20]) i znany jest dokładny algorytm wielomianowy, metoda sympleks w większości przypadków działa szybciej, stąd jest najczęściej stosowana w praktyce.

Pewną modyfikacją problemu programowania liniowego jest wprowadzenie ograniczenia, pozwalającego zmiennym decyzyjnym na przyjmowanie wyłącznie wartości całkowitych. Problem taki nazywany jest programowaniem liniowym *całkowitoliczbowym*, lub *mieszanym całkowitoliczbowym* (gdy nie wszystkie zmienne muszą być całkowite). Wprowadzając dodatkowe ograniczenie mówiące, że zmienne decyzyjne nie mogą być większe od 1, formułujemy zadanie programowania *binarnego* (*zerojedynkowego*). Ma ono liczne zastosowania, z uwagi na fakt, że za pomocą zmiennych binarnych można wyrazić wiele typowych problemów kombinatorycznych. Niestety, wprowadzając zmienne całkowite, powodujemy drastyczne zwiększenie kosztów obliczeń - problem staje się NP-trudny i nierozwiązywalny przez żaden znany algorytm wielomianowy. Istnieją jednak metody radzenia sobie z problemem w sposób wykładniczy (np. metoda płaszczyzn odcinających Gomorego) lub heurystyczny. Szersze omówienie powyższych zagadnień znajduje się w [1] i [12], podstaw matematycznych można szukać w podręcznikach od algebry liniowej, np. [28].

W przypadku regresji zbioru danych za pomocą funkcji odcinkami liniowej problem ten da się wyrazić za pomocą zadania programowania liniowego. Wzorowano się tutaj na rozwiązaniu zastosowanym w jednej z metod wspomagania decyzji, nazwanej UTA (*Utilité Additive*) [21]. Zastosowano tam funkcję użyteczności będącą sumą funkcji cząstkowych odcinkami liniowych, monotonicznych, mając dany porządek na podzbiorze wariantów decyzyjnych, stworzony przez decydenta i użyto do jej konstrukcji właśnie programowania liniowego. W naszym przypadku mamy do czynienia z problemem podobnym, występują jednak dodatkowe założenia nakazujące, aby wartość funkcji dolnego przybliżenia w punktach, gdzie znajdują się obiekty, była mniejsza od wartości dolnego przybliżenia zbioru decyzyjnego, dla funkcji górnego przybliżenia — większa. Dodatkowo, w sekcji 5.3.3 opisano sposób regresji funkcji w przypadku, gdy nie jest znana liczba odcinków funkcji odcinkami liniowych, wykorzystując przy tym programowanie całkowitoliczbowe mieszane. Programem, za pomocą którego rozwiązano powyższe problemy, jest GAMS[10], profesjonalny, płatny zestaw *solverów*, rozwiązujących zagadnienia programowania matematycznego, zarówno liniowego, jak i nieliniowego. Przedtem używana była darmowa biblioteka o otwartym kodzie *lp-solve*[24] autorstwa Michiela Berkelaara.

5.3.2 Przypadek regresji przy zadanych punktach charakterystycznych funkcji

Rozważmy zbiór n obiektów $Q = \{x_1, x_2, \dots, x_n\}$, opisanych za pomocą m kryteriów i atrybutu decyzyjnego. Wartość i -tego obiektu na j -tym kryterium oznaczmy przez $q_j(x_i)$ i będzie ona odpowiadała wartości funkcji przynależności do zbioru rozmytego X_j opartego na j -tym atrybucie, tzn. $\mu_{X_j}(x_i)$; wartość atrybutu decyzyjnego oznaczmy przez $dec(x_i)$ i odpowiada ona wiarygodności decyzji danego obiektu, tzn. $\mu_Y(x_i)$. Dla uproszczenia założymy, że wszystkie kryteria są typu *zysk*.

Poprzez zastosowanie metod teorii zbiorów przybliżonych w ramach podejścia opartego na regułach gradualnych, opisanego w poprzednich rozdziałach, każdemu obiektowi można przypisać przynależność do zbiorów rozmytych określających górne i dolne przybliżenie zbioru Y (decyzji), oznaczane odpowiednio jako $\mu[\overline{App}(X_1, \dots, X_n, Y)]$ oraz $\mu[\underline{App}(X_1, \dots, X_n, Y)]$. Dla uproszczenia notacji przynależność do zbiorów górnego i dolnego przybliżenia będzie w tym rozdziale oznaczana jako $\overline{dec}(x_i)$ i $\underline{dec}(x_i)$. W ogólności, naszym celem będzie próba aproksymowania dwóch funkcji, które ograniczą zbiór obiektów od dołu i od góry w przestrzeni kryteriów; funkcje oznaczmy przez $f(\cdot)$ i $g(\cdot)$. Przyjmijmy, że pierwsza z nich w punktach, w których znajdują się obiekty, powinna przyjmować wartości równe dolnym przybliżeniom ich decyzji, a druga z nich — górnym przybliżeniom decyzji; stąd w przypadku, w którym nie będzie błędu przy aproksymacji otrzymujemy:

$$f(x_i) = \underline{dec}(x_i) \quad (5.9)$$

$$g(x_i) = \overline{dec}(x_i) \quad (5.10)$$

Problem sprowadza się do regresji obu funkcji, to znaczy wyznaczenia ich wartości we wszystkich punktach przestrzeni. Muszą zostać przy tym spełnione następujące warunki:

1. Funkcje muszą aproksymować zbiór w sposób spójny — dla żadnej pary punktów w przestrzeni kryteriów nie może zaistnieć taka sytuacja, że pierwszy z punktów będzie dominował drugi, a zarazem wartość funkcji w pierwszym z punktów będzie mniejsza od wartości w drugim z nich. Jest to niezgodne z zasadą dominacji, będącą podstawą podejścia opartego na regułach gradualnych. Można wykazać, że funkcja taka musi być monotoniczna ze względu na każde kryterium.
2. Funkcja $f(\cdot)$ musi ograniczać zbiór obiektów od dołu, stąd w punktach, w których znajdują się obiekty, może przyjąć wartość nie większą od wartości dolnego przybliżenia decyzji obiektów. Funkcja $g(\cdot)$ musi ograniczać zbiór obiektów od góry, stąd może przyjąć wartość nie mniejszą od wartości górnego przybliżenia decyzji obiektów.
3. W każdym punkcie przestrzeni musi zachodzić $f(\alpha) \leq g(\alpha)$

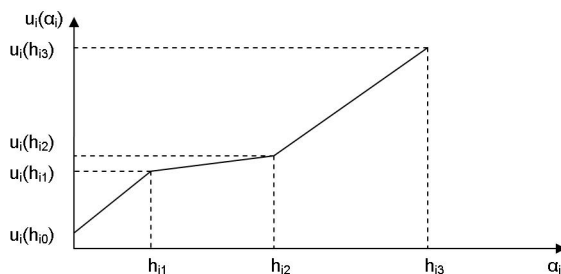
Rozważmy funkcje $f(\cdot)$ i $g(\cdot)$ w postaci sumy funkcji cząstkowych na poszczególnych kryteriach:

$$f(\alpha) = \sum_{j=1}^m u_j(q_j(\alpha)) + u_0 \quad (5.11)$$

$$g(\alpha) = \sum_{j=1}^m v_j(q_j(\alpha)) + v_0 \quad (5.12)$$

Przyjmujemy, że funkcje cząstkowe $u_j(\cdot)$ i $v_j(\cdot)$ są odcinkami liniowymi, przy czym miejsca (wartości), w których funkcje zmieniają współczynnik nachylenia są z góry ustalone (nie są zmiennymi, ale stałymi parametrami problemu), zarówno ich położenie, jak również ich liczba; miejsca takie będziemy od teraz nazywali *punktami złamań*. Przykładowy kształt funkcji $u_j(\cdot)$ został przedstawiony na rysunku 5.3. Oznaczając przez $h_{i,j}^u$ wartość i -tego punktu złamania na j -tym kryterium dla tej funkcji można zapisać wartość $u_j(\cdot)$ między punktami $h_{i,j}^u$ a $h_{i+1,j}^u$ jako:

$$u_j(q_j(x)) = u_j(h_{i,j}^u) + (q_j(x) - h_{i,j}^u) \cdot \frac{u_j(h_{i+1,j}^u) - u_j(h_{i,j}^u)}{h_{i+1,j}^u - h_{i,j}^u} \quad (5.13)$$



Rysunek 5.3: Przykład funkcji cząstkowej odcinkami liniowej

Oznaczmy liczbę punktów złamań dla funkcji $u_j(\cdot)$ i $v_j(\cdot)$ odpowiednio przez s_j^u i s_j^v . Warunkiem zapewniającym monotoniczność na każdym kryterium będzie nierówność:

$$u_j(h_{i,j}^u) \geq u_j(h_{i,j}^u) \quad \text{dla } 1 \leq i \leq s_j^u, 1 \leq j \leq m \quad (5.14)$$

$$v_j(h_{i+1,j}^v) \geq v_j(h_{i,j}^v) \quad \text{dla } 1 \leq i \leq s_j^v, 1 \leq j \leq m \quad (5.15)$$

Z kolei warunkiem, który zapewni, że funkcje $f(x_i)$ i $g(x_i)$ się nie przetną, będzie:

$$u_j(h_{i,j}^u) \geq v_j(h_{k,j}^v) \quad \text{dla } 1 \leq i \leq s_j^u, 1 \leq k \leq s_j^v, 1 \leq j \leq m \quad (5.16)$$

Warunki na wartości funkcji dla obiektów z tablicy decyzyjnej powinny być następujące:

$$f(x_i) + \sigma_i^f = \underline{dec}(x_i) \quad \text{dla } 1 \leq i \leq n \quad (5.17)$$

$$g(x_i) - \sigma_i^g = \overline{dec}(x_i) \quad \text{dla } 1 \leq i \leq n \quad (5.18)$$

gdzie σ_i^f oraz σ_i^g są to błędy aproksymacji, które muszą zostać uwzględnione, ponieważ nie zawsze będzie możliwe bezbłędne aproksymowanie przykładów z tablicy decyzyjnej za pomocą konstruowanej funkcji. Zauważmy, że skoro oba błędy zawsze są dodatnie (programowanie liniowe nakazuje narzucić zmiennym takie ograniczenie), to z powodu dodania błędu, funkcja $f(x_i)$ będzie zawsze mniejsza od $\underline{dec}(x_i)$, natomiast funkcja $g(x_i)$, z powodu odjęcia błędu, będzie zawsze większa od $\overline{dec}(x_i)$. W późniejszym rozwiązaniu korzystnie jest dołożyć do tablicy decyzyjnej dwa sztuczne obiekty — jeden o wszystkich współrzędnych i decyzji równych zeru (wartości minimalnej), drugi — równych jedności (wartości maksymalnej). Pozwala to na kontrolę przebiegu funkcji na kresach dziedziny.

Reasumując, pełen problem programowania liniowego możemy zapisać w postaci:

$$\min z = \sum_{i=1}^n (\sigma_i^f + \sigma_i^g) \quad (5.19)$$

$$\text{p.o.} \quad (5.20)$$

$$\sum_{j=1}^m u_j(q_j(x_i)) + u_0 + \sigma_i^f = \underline{dec}(x_i) \quad \text{dla } 1 \leq i \leq n \quad (5.21)$$

$$\sum_{j=1}^m v_j(q_j(x_i)) + v_0 - \sigma_i^g = \overline{dec}(x_i) \quad \text{dla } 1 \leq i \leq n \quad (5.22)$$

$$u_j(h_{i,j}^u) \geq u_j(h_{i,j}^u) \quad \text{dla } 1 \leq i \leq s_j^u, 1 \leq j \leq m \quad (5.23)$$

$$v_j(h_{i+1,j}^v) \geq v_j(h_{i,j}^v) \quad \text{dla } 1 \leq i \leq s_j^v, 1 \leq j \leq m \quad (5.24)$$

$$u_j(h_{i,j}^u) \geq v_j(h_{k,j}^v) \quad \text{dla } 1 \leq i \leq s_j^u, 1 \leq k \leq s_j^v, 1 \leq j \leq m \quad (5.25)$$

Oczywiście, wszystkie zmienne decyzyjne muszą być nieujemne, ale jest to domyślne założenie programowania liniowego, stąd dla przejrzystości nie zostało tutaj wypisane. Dodajmy też, że wszystko, co tutaj rozważaliśmy, odnosi się do kryteriów typu *zysk*. Dla kryteriów typu *koszt* znaki w równaniach (5.23) i (5.24) musiałyby zostać zamienione na przeciwne.

5.3.3 Przypadek regresji dla punktów charakterystycznych funkcji traktowanych jako zmienne decyzyjne

Rozważmy teraz problem, w którym nie ustalamy z góry, na ile odcinków zostanie podzielona każda z funkcji cząstkowych. Sytuacja powyższa korzystna jest przede wszystkim dlatego, że zależy nam na jak najprostszym opisie zbioru. Skoro udałoby się zbiór opisać za pomocą funkcji liniowych (bez punktów złamań), lub za pomocą funkcji cząstkowych z jednym lub dwoma punktami złamania, nie popełniając dużego błędu, to czy jest sens dodawać dodatkowe punkty, tworzyć skomplikowany model wyłącznie po to, aby zmniejszyć błąd o niedostrzegalną w praktyce wartość? Rozważany problem odpowiada zjawisku *przeuczenia się* występującemu w uczeniu indukcyjnym, polegającemu na nadmiernym dopasowaniu modelu do danych przez algorytm uczący. Model okazuje się być całkowicie spójny ze zbiorem lub błąd na zbiorze jest bardzo mały, ale jest zarazem sztucznie skomplikowany. Przycinanie drzew decyzyjnych czy zbiorów reguł ma na celu jego uproszczenie, mimo spadku jakości klasyfikacji. Model prostszy ma większą szansę być wiarygodnym i poprawnym nie tylko na zbiorze uczącym, ale dla wszystkich późniejszych obiektów (brzytwa Ockhama); dodatkowo zdecydowanie ułatwia wyjaśnienia zbioru, czyli ujęcie w sposób zrozumiały dla człowieka zależności tkwiących w danych.

Algorytm przedstawiony w poprzedniej sekcji stara się jak najdokładniej opisać zbiór przy ustalonej liczbie punktów złamań. Jeśli w trakcie rozwiązywania problemu punkty takie — ich liczba i umiejscowienie — zostałyby dobrane przez algorytm, pozwoliłoby to, mimo popełniania większego błędu, na uproszczenie modelu a zarazem wykrycie specyficznych miejsc zmiany trendu. Rozpocniemy od podzielenia każdego z kryteriów na szereg odcinków (czyli postawienia na osi każdego z kryteriów pewnej ilości punktów), co w praktyce wykonać w następujący sposób:

- dzieląc taką oś (ograniczoną z dołu zerem i z góry jednością) na odcinki o tej samej długości,
- dzieląc oś na odcinki zgodnie z częstością występowania przykładów uczących — duże ich zagęszczenie powoduje gęstsze podziały, najłatwiej podzielić oś tak, aby na każdym odcinku znalazło się tyle samo przykładów uczących,
- w szczególności przy liczbie punktów równej liczbie przykładów (obiektów), położenie punktów jest określone przez współrzędne obiektów na danym kryterium.

Niezależnie od wybranego sposobu przyjmijmy, że udało nam się na każdym z kryteriów ustawić pewną liczbę punktów, które są teraz potencjalnymi punktami złamań. Problem, który należy rozwiązać można sformułować następująco: spośród wszystkich dostępnych punktów wybrać jak najmniejszą ich liczbę oraz określić w nich jak najlepsze współczynniki kolejnych odcinków funkcji liniowej tak, aby błąd aproksymacji zbioru był jak najmniejszy. Mamy w tym

momencie problem dwukryterialnej optymalizacji: musimy jednocześnie minimalizować błąd funkcji na zbiorze danych oraz ilość punktów złamań. Są to kryteria w ogólności przeciwstawne sobie, więc do kompromisu może dojść na drodze przetargu. Poniżej zaprezentowano najprostszy sposób rozwiązania problemu: minimalizację jednego kryterium przy zadaniu pewnego proggu, którego drugie z kryteriów nie może przekroczyć.

Ponownie rozważmy zbiór n obiektów $Q = \{x_1, x_2, \dots, x_n\}$, opisanych za pomocą m kryteriów i wartości wiarygodności decyzji. Pozostawimy oznaczenie $\overline{dec}(x_i)$ na górne przybliżenie zbioru decyzji $\mu[\overline{App}(X_1, \dots, X_n, Y)]$ oraz oznaczenie $\underline{dec}(x_i)$ na dolne przybliżenie $\mu[\underline{App}(X_1, \dots, X_n, Y)]$. Ponownie rozważymy dwie funkcje, $f(\cdot)$ i $g(\cdot)$, składające się z funkcji cząstkowych odcinkami liniowymi:

$$f(\alpha) = \sum_{j=1}^m u_j(q_j(\alpha)) + u_0 \quad (5.26)$$

$$g(\alpha) = \sum_{j=1}^m v_j(q_j(\alpha)) + v_0 \quad (5.27)$$

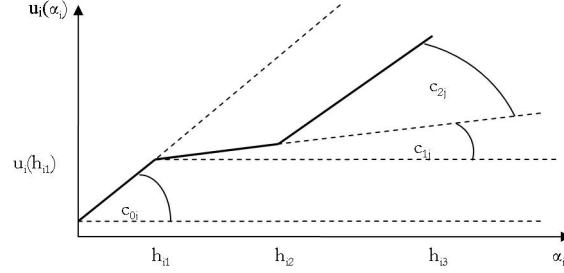
Musimy jednak przedefiniować funkcje cząstkowe. Użyjemy innego zapisu dla funkcji odcinkami liniowej: przyjmijmy, że funkcja cząstkowa $u_j(\cdot)$ może zostać złamana potencjalnie w s_j^u punktach, przy czym każdemu takiemu punktowi odpowiada punkt na osi j -tego kryterium o współrzędnej $h_{i,j}$. Jeśli przez $c_{0,j}$ oznaczymy współczynnik kierunkowy (nachylenie) pierwszego z odcinków takiej funkcji, a przez $c_{i,j}$ zmianę współczynnika kierunkowego takiej funkcji w i -tym punkcie złamania, to wartość funkcji dla punktu leżącego między punktami $h_{i,j}$ a $h_{i+1,j}$ można zapisać jako (patrz rysunek 5.4):

$$u_j(q_j(\alpha)) = c_{0,j}q_j(\alpha) + \sum_{k=1}^i c_{k,j}(q_j(\alpha) - h_{k,j}) \quad (5.28)$$

$$v_j(q_j(\alpha)) = d_{0,j}q_j(\alpha) + \sum_{k=1}^i d_{k,j}(q_j(\alpha) - h_{k,j}) \quad (5.29)$$

gdzie $c_{k,j}$ są współczynnikami kierunkowymi funkcji $f(\cdot)$, a $d_{k,j}$ — funkcji $g(\cdot)$.

Ponieważ współczynnik kierunkowy może być dodatni albo ujemny, a w problemie programowani liniowego mamy do czynienia tylko ze zmiennymi nieujemnymi, możemy dokonać zmiany w powyższym równaniu poprzez zastąpienie każdego współczynnika $c_{k,j}$ różnicą współczynników $c_{k,j}^+ - c_{k,j}^-$, gdzie $c_{k,j}^+$ określa dodatnią zmianę współczynnika kierunkowego w k -tym punkcie złamania, natomiast $c_{k,j}^-$ — zmianę ujemną (analogicznie dla $d_{k,j}$). Aby zapewnić monotoniczność funkcji $f(\cdot)$ i $g(\cdot)$ ze względu na każde kryterium, należy zapewnić monotoniczność każdej z funkcji cząstkowych. Tym samym należy zapewnić, że współczynnik kierunkowy takiej funkcji cząstkowej w każdym punkcie będzie nieujemny. Stąd suma wszystkich zmian współczynnika dla każdego punktu złamania i punktów poprzedzających go musi być nieujemna, czyli:



Rysunek 5.4: Interpretacja nowego zapisu funkcji cząstkowej odcinkami liniowej

$$\sum_{k=0}^i (c_{k,j}^+ - c_{k,j}^-) \geq 0 \quad \text{dla } 0 \leq i \leq s_j^u, 1 \leq j \leq m \quad (5.30)$$

$$\sum_{k=0}^i (d_{k,j}^+ - d_{k,j}^-) \geq 0 \quad \text{dla } 0 \leq i \leq s_j^v, 1 \leq j \leq m \quad (5.31)$$

Czas wprowadzić zmienne kontrolujące ilość punktów złamań. Oznaczmy przez $\chi_{i,j}$ zmienną binarną, która przyjmuje wartość jeden, gdy dla funkcji cząstkowej $u_j(\cdot)$ w i -tym potencjalnym punkcie złamania faktycznie ono nastąpi. W przeciwnym przypadku (brak złamania w tym punkcie) zmienna przyjmie wartość zero. Podobnie, zdefiniujemy taką samą zmienną binarną $\phi_{i,j}$ dla funkcji cząstkowych $v_j(\cdot)$. Pozwala to na zapis funkcji, które łamią się wyłącznie w części z potencjalnych punktów w następujący sposób (zostawiono współczynniki $c_{k,j}$ zamiast $c_{k,j}^+ - c_{k,j}^-$ w celu poprawy przejrzystości):

$$u_j(q_j(\alpha)) = c_{0,j} q_j(\alpha) + \sum_{k=1}^i \chi_{k,j} c_{k,j} (q_j(\alpha) - h_{k,j}) \quad (5.32)$$

$$v_j(q_j(\alpha)) = d_{0,j} q_j(\alpha) + \sum_{k=1}^i \phi_{k,j} d_{k,j} (q_j(\alpha) - h_{k,j}) \quad (5.33)$$

Wprowadza to możliwość manewrowania miejscami złamań jako zmiennymi, a nie ustalonymi parametrami. Pojawia się tu jednak czynnik nieliniowy (iloczyn dwóch zmiennych), co znacznie zwiększa koszt algorytmu, wystarczająco już wysoki z powodu użycia zmiennych całkowitych. Pokażemy, że można powyższy czynnik wyleminować. Mianowicie, równania (5.32) i (5.33) zastępujemy z powrotem równaniami (5.28) i (5.29), oraz dodajemy dwa kolejne równania:

$$c_{i,j}^+ + c_{i,j}^- \leq \Gamma \cdot \chi_{i,j} \quad (5.34)$$

$$d_{i,j}^+ + d_{i,j}^- \leq \Gamma \cdot \phi_{i,j} \quad (5.35)$$

gdzie Γ to pewna ustalona liczba, na tyle duża, że w realnych problemach żadna zmiana współczynnika kierunkowego jej nie przekroczy. Wydaje się, że w przypadku nie za gęstego ułożenia potencjalnych punktów złamań, wystarczy przyjąć $\Gamma = 10$, gdyż takie nachylenie oznacza skok wartości decyzji o 1 (czyli od minimum do maksimum) na odcinku 0.1. Nic nie stoi na przeszkodzie, aby przyjąć bardziej bezpieczną wartość, np. $\Gamma = 100$. Analizując powyższe równania, można zauważyć, że jeśli $\chi_{i,j}$ (lub $\phi_{i,j}$) jest równa 0, współczynniki $c_{i,j}^+$ i $c_{i,j}^-$ (odpowiednio $d_{i,j}^+$ i $d_{i,j}^-$) muszą być zerami (nie ma w tym punkcie złamania). Jeśli natomiast $\chi_{i,j}$ (lub $\phi_{i,j}$) jest równe 1, z uwagi na dużą wartość Γ współczynniki kierunkowe mogą przyjąć praktycznie dowolną wartość. Jest to właśnie zagadnienie, który chcieliśmy rozwiązać i udało się to zrobić nie wprowadzając do równań czynników nieliniowych.

Występuje tu jeszcze jeden problem, dotyczący samego algorytmu sympleks. W sekcji 5.3.1 wspomniano, że algorytm przeszukuje jedynie wierzchołki wielościanu wypukłego rozwiązań dopuszczalnych, co powoduje często przyjmowanie przez zmienne wartości skrajnych, o ile są dopuszczalne i optymalne. W praktyce może zdarzyć się, że, jeśli nie zakłóci to optymalności rozwiązania, współczynniki kierunkowe przyjmą wartości możliwe największe, czyli równe Γ . Jest to niemiłe widziane choćby dlatego, że wartość Γ jest *dużą liczbą* i nie powinna być kojarzona z żadną konkretną wartością, a przynajmniej wielkości współczynników nie powinny zależeć od tego, jak duże Γ zostanie przyjęte. Po drugie, wielkie współczynniki nie mają w tym problemie głębszego sensu. Dlatego będziemy starali się minimalizować również same współczynniki przy zachowaniu optymalności rozwiązania. Robimy to poprzez jawne wprowadzenie tzw. zmiennych osłabiających (ang. *slack variables*). Są one bez wiedzy użytkownika stosowane przez *solvery* przy rozwiązywaniu problemów, nam zależy jednak na tym, aby je maksymalizować, stąd musimy je włączyć do naszych rozważań:

$$c_{i,j}^+ + c_{i,j}^- + \eta_{i,j}^f = \Gamma \cdot \chi_{i,j} \quad (5.36)$$

$$d_{i,j}^+ + d_{i,j}^- + \eta_{i,j}^g = \Gamma \cdot \phi_{i,j} \quad (5.37)$$

gdzie $\eta_{i,j}^f$ i $\eta_{i,j}^g$ to właśnie wspomniane zmienne osłabiające. Będą one maksymalizowane w funkcji celu, ale z bardzo małą wagą. Spowoduje to przyjęcie jak najmniejszych współczynników kierunkowych, pozostawiając rozwiązanie optymalnym — mała waga ma na celu optymalizację leksykograficzną: wpraw minimalizujemy same błędy (lub ilość punktów złamań), a dopiero w ramach znalezionej optymalności próbujemy minimalizować współczynniki. Można to osiągnąć, wprowadzając *bardzo dużą liczbę* M równą np. 1000 i minimalizując z taką wagą wszystkie zmienne występujące w funkcji celu z wyjątkiem zmiennych osłabiających.

Dochodzimy tutaj do problemu, co w gruncie rzeczy ma być optymalizowane — liczba punktów złamań czy błędy aproksymacji? Zdecydowano się na pierwszą z tych opcji, gdyż jest, przynajmniej zdaniem autora, bardziej intuicyjna. Wydaje się, że łatwiej jest podać maksymalną wartość błędu, jakiej

nie może przekroczyć funkcja i wyznaczyć w ten sposób minimalną liczbę złamań, niż na odwrót. Jest to jednak problem jawnie wielokryterialny i nie ma podejścia najlepszego — każde jest tylko kompromisem (jeśli zależy nam na jak dokładniejszym szacowaniu punktów, możemy pozostać przy minimalizacji błędu). Trzymając się jednak przyjętych założeń, przyjmijmy maksymalną wartość błędu funkcji górnego i dolnego przybliżenia jako δ . Dodatkowo, dla uproszczenia założymy (zupełnie rozsądnie) tą samą liczbę i te same miejsca potencjalnych punktów złamań dla funkcji $u_j(\cdot)$ i $v_j(\cdot)$, czyli $h_{i,j}^u = h_{i,j}^v = h_{i,j}$ oraz $s_j^u = s_j^v = s_j$. W tym momencie mamy już wszystkie potrzebne dane i ograniczenia do zapisania rozważanego problemu jako zagadnienia programowania liniowego całkowitoliczbowego mieszanego:

$$\min z = M \cdot \sum_{j=1}^m \sum_{i=1}^{s_j} (\chi_{i,j} + \phi_{i,j}) - \sum_{j=1}^m \sum_{i=1}^{s_j} (\eta_{i,j}^f + \eta_{i,j}^g) \quad (5.38)$$

$$\text{p.o.} \quad (5.39)$$

$$\sum_{j=1}^m u_j(q_j(x_i)) + u_0 + \sigma_i^f = \underline{dec}(x_i) \quad \text{dla } 1 \leq i \leq n \quad (5.40)$$

$$\sum_{j=1}^m v_j(q_j(x_i)) + v_0 - \sigma_i^g = \overline{dec}(x_i) \quad \text{dla } 1 \leq i \leq n \quad (5.41)$$

$$u_j(h_{i,j}^u) \geq v_j(h_{i,j}^u) \quad \text{dla } 1 \leq i \leq s_j, 1 \leq j \leq m \quad (5.42)$$

$$c_{i,j}^+ + c_{i,j}^- + \eta_{i,j}^f = \Gamma \cdot \chi_{i,j} \quad \text{dla } 1 \leq i \leq s_j, 1 \leq j \leq m \quad (5.43)$$

$$d_{i,j}^+ + d_{i,j}^- + \eta_{i,j}^g = \Gamma \cdot \phi_{i,j} \quad \text{dla } 1 \leq i \leq s_j, 1 \leq j \leq m \quad (5.44)$$

$$\sum_{k=0}^i (c_{k,j}^+ - c_{k,j}^-) \geq 0 \quad \text{dla } 0 \leq i \leq s_j, 1 \leq j \leq m \quad (5.45)$$

$$\sum_{k=0}^i (d_{k,j}^+ - d_{k,j}^-) \geq 0 \quad \text{dla } 0 \leq i \leq s_j, 1 \leq j \leq m \quad (5.46)$$

$$\sum_{i=1}^n (\sigma_i^f + \sigma_i^g) \leq \delta \quad (5.47)$$

gdzie za $u_j(q_j(\alpha))$ i $v_j(q_j(\alpha))$ podstawiamy:

$$u_j(q_j(\alpha)) = (c_{0,j}^+ - c_{0,j}^-)q_j(\alpha) + \sum_{k:h_{k,j} \leq \alpha} (c_{k,j}^+ - c_{k,j}^-)(q_j(\alpha) - h_{k,j}) \quad (5.48)$$

$$v_j(q_j(\alpha)) = (d_{0,j}^+ - d_{0,j}^-)q_j(\alpha) + \sum_{k:h_{k,j} \leq \alpha} (d_{k,j}^+ - d_{k,j}^-)(q_j(\alpha) - h_{k,j}) \quad (5.49)$$

Do sformułowanego problemu dodamy jeszcze kilka komentarzy:

- w funkcji celu w równaniu (5.38) zmienne osłabiające są *maksymalizowane*, więc druga suma wzięta jest ze znakiem ujemnym.
- w równaniach (5.48) i (5.49) sumy wzięte są po wszystkich współczynnikach $c_{k,j}^+, c_{k,j}^-, d_{k,j}^+, d_{k,j}^-$ dla których odpowiadające im punkty $h_{k,j}$ są mniejsze od argumentu funkcji α
- liczba punktów złamań równa jest liczbie odcinków, na jakie jest podzielona funkcja *minus jeden*; współczynniki w punkcie zerowym (tzn. $c_{0,j}^+, c_{0,j}^-, d_{0,j}^+, d_{0,j}^-$) są różne od zera, nie ma zmiennych binarnych odpowiedzialnych za złamania w tych punktach. Stąd otrzymujemy funkcję liniową, gdy liczba punktów złamań jest równa zero.
- suma w wyrażeniu określającym funkcję celu powinna być wielokrotnością M , ale nie będzie, ponieważ zmniejszy ją trochę suma zmiennych osłabiających. Aby otrzymać minimalną liczbę punktów złamań, można funkcję celu podzielić przez M i przyrównać do najbliższej większej liczby całkowitej.
- Rozważany problem odnosi się do kryteriów typu *zysk*. Dla kryteriów typu *koszt* znaki w równaniach (5.45) i (5.46) musiałyby zostać zamienione na przeciwne.

Na koniec przypomnijmy, że problem jest zagadnieniem programowania liniowego całkowitoliczbowego mieszanego. Oznacza to wysoką złożoność obliczeniową i potrzebę użycia heurystyk lub wykładniczych, ale w praktyce efektywnych algorytmów (np. Gomory’ego). W wymienione algorytmy wyposażone są *solvery*, rozwiązujące tego typu zagadnienia. Powoduje to separację samego sformułowania problemu od sposobu jego rozwiązania — drugą z tych czynności wykonują zoptymalizowane programy. Należy sobie jednak zdawać sprawę, że między programowaniem liniowym, a nieliniowym istnieje ogromna różnica w stopniu trudności, stąd tyle wysiłku, aby uniknąć nieliniowości w ograniczeniach. Aby jeszcze skrócić obliczenia, należałoby jak najbardziej zmniejszyć liczbę zmiennych całkowitych. Można tego dokonać poprzez redukcję liczby kryteriów, a konkretnie poprzez użycie reduktu zamiast pełnego ich zbioru. Nie ma to wpływu na jakość klasyfikacji, a znacznie polepsza szybkość działania algorytmu.

5.3.4 Przykład regresji

Aby zilustrować działanie metod, na przykładowym zbiorze, który był już używany w poprzednich rozdziałach, dokonamy regresji dwoma sposobami (z ustaloną liczbą punktów złamań oraz z liczbą złamań jako zmienną w problemie). Pozwolimy sobie przypomnieć, jak wygląda zbiór danych, wraz z wiarygodnościami dolnego i górnego przybliżenia, zamieszczając go w tabeli 5.3.

Rozpocznijmy od ustalonych punktów złamań, czyli zwykłego programowania liniowego. Przyjmijmy, że złamanie może nastąpić dla każdej wartości na danym kryterium, która występuje w tablicy decyzyjnej, czyli dla X_1

Nr	$q_1(x)$	$q_2(x)$	$dec(x)$	$\underline{dec}(x)$	$\overline{dec}(x)$
1	0.3	0.1	0.1	0.1	0.1
2	0.4	0.3	0.2	0.2	0.25
3	0.2	0.3	0.25	0.2	0.25
4	0.6	0.5	0.3	0.3	0.7
5	0.8	0.15	0.5	0.5	0.5
6	0.7	0.45	0.6	0.6	0.7
7	0.3	0.6	0.65	0.55	0.65
8	0.85	0.8	0.75	0.75	0.75
9	0.35	0.6	0.55	0.55	0.65
10	0.55	0.4	0.7	0.3	0.7

Tablica 5.3: Przykładowy zbiór danych.

będą to punkty: 0.2, 0.3, 0.35, 0.4, 0.55, 0.6, 0.7, 0.8, 0.85, natomiast dla X_2 — 0.1, 0.15, 0.3, 0.4, 0.45, 0.5, 0.6, 0.8. Tym samym nastawiamy się na znalezienie funkcji z jak najmniejszym błędem. Przypomnijmy, że $f(x) = u_0 + u_1(q_1(x)) + u_2(q_2(x))$, oraz $g(x) = v_0 + v_1(q_1(x)) + v_2(q_2(x))$.

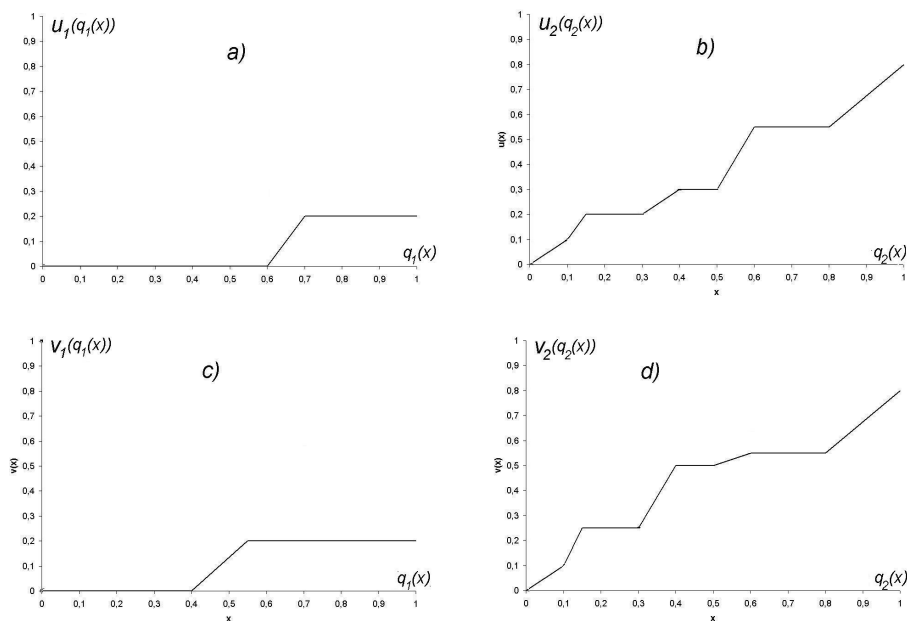
W wyniku otrzymujemy $u_0 = v_0 = 0$ oraz:

$$u_1(q_1(x)) = \begin{cases} 0 & q_1(x) < 0.6 \\ 2q_1(x) - 1.2 & 0.6 \leq q_1(x) < 0.7 \\ 0.2 & q_1(x) \geq 0.7 \end{cases} \quad (5.50)$$

$$u_2(q_2(x)) = \begin{cases} q_2(x) & q_2(x) < 0.1 \\ 2q_2(x) - 0.1 & 0.1 \leq q_2(x) < 0.15 \\ 0.2 & 0.15 \leq q_2(x) < 0.3 \\ q_2(x) - 0.1 & 0.3 \leq q_2(x) < 0.4 \\ 0.3 & 0.4 \leq q_2(x) < 0.5 \\ 2.5q_2(x) - 0.95 & 0.5 \leq q_2(x) < 0.6 \\ 0.55 & 0.6 \leq q_2(x) < 0.8 \\ 1.25q_2(x) - 0.45 & q_2(x) \geq 0.8 \end{cases} \quad (5.51)$$

$$v_1(q_1(x)) = \begin{cases} 0 & q_1(x) < 0.4 \\ 1.33q_1(x) - 0.53 & 0.4 \leq q_1(x) < 0.55 \\ 0.2 & q_1(x) \geq 0.55 \end{cases} \quad (5.52)$$

$$v_2(q_2(x)) = \begin{cases} q_2(x) & q_2(x) < 0.1 \\ 3q_2(x) - 0.2 & 0.1 \leq q_2(x) < 0.15 \\ 0.25 & 0.15 \leq q_2(x) < 0.3 \\ 2.5q_2(x) - 0.5 & 0.3 \leq q_2(x) < 0.4 \\ 0.5 & 0.4 \leq q_2(x) < 0.5 \\ 0.5q_2(x) + 0.25 & 0.5 \leq q_2(x) < 0.6 \\ 0.55 & 0.6 \leq q_2(x) < 0.8 \\ 1.25q_2(x) - 0.45 & q_2(x) \geq 0.8 \end{cases} \quad (5.53)$$



Rysunek 5.5: Funkcje aproksymujące zbiór dla ustalonej liczby punktów złamań (programowanie liniowe): $u_1(q_1(x))$ (a), $u_2(q_2(x))$ (b), $v_1(q_1(x))$ (c), $v_2(q_2(x))$ (d)

Funkcje powyższe zostały również przedstawione na rysunku 5.5. Widać, że funkcja dla drugiego z kryteriów (X_2) jest nadmiernie skomplikowana i prawdopodobnie zbyt mocno dopasowana do danych.

W związku z tym spróbujemy użyć drugiej z metod — poszukamy optymalnej liczby i lokalizacji punktów złamań przy błędzie nie przekraczającym pewnej wartości. Wydaje się rozsądne ustalić, że średni błąd przypadający na obiekt nie powinien być większy od 0.05. Tym samym całkowity błąd nie powinien przekroczyć 1.2 (dwie funkcje — $f(\cdot)$ i $g(\cdot)$, po dziesięć obiektów oraz punkty $[0, 0]$ i $[1, 1]$, wszędzie średni błąd 0.05).

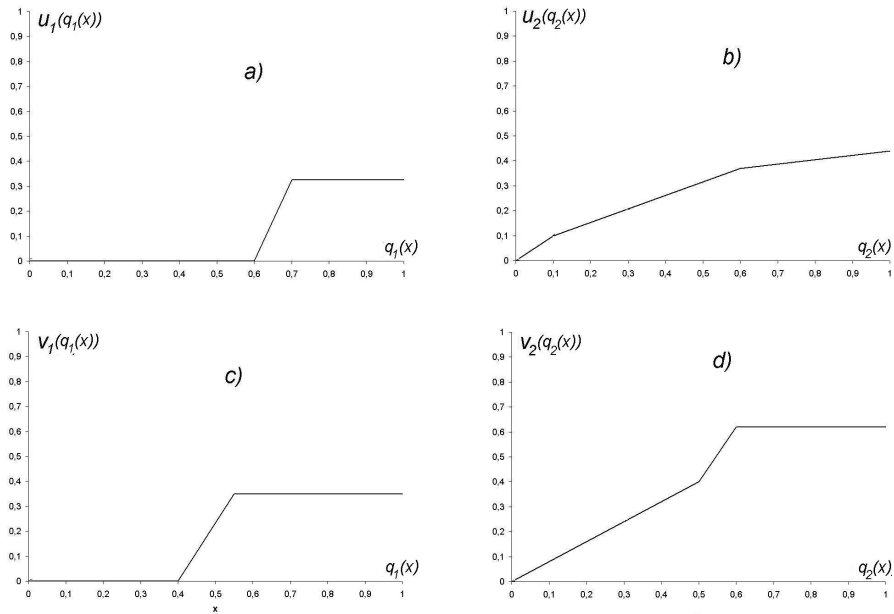
Rozwiązując powyższy problem za pomocą programowania liniowego całkowitoliczbowego mieszanego, otrzymujemy $u_0 = 0$, $v_0 = 0.027$, oraz:

$$u_1(q_1(x)) = \begin{cases} 0 & q_1(x) < 0.6 \\ 3.25q_1(x) - 1.95 & 0.6 \leq q_1(x) < 0.7 \\ 0.18q_1(x) + 0.2 & q_1(x) \geq 0.7 \end{cases} \quad (5.54)$$

$$u_2(q_2(x)) = \begin{cases} q_2(x) & q_2(x) < 0.1 \\ 0.5q_2(x) + 0.05 & 0.1 \leq q_2(x) < 0.6 \\ 0.24q_2(x) + 0.2 & q_2(x) \geq 0.6 \end{cases} \quad (5.55)$$

$$v_1(q_1(x)) = \begin{cases} 0 & q_1(x) < 0.4 \\ 2.36q_1(x) - 0.94 & 0.4 \leq q_1(x) < 0.55 \\ 0.35 & q_1(x) \geq 0.55 \end{cases} \quad (5.56)$$

$$v_2(q_2(x)) = \begin{cases} 0.8q_2(x) & q_2(x) < 0.5 \\ 2.23q_2(x) - 0.72 & 0.5 \leq q_2(x) < 0.6 \\ 0.62 & q_2(x) \geq 0.6 \end{cases} \quad (5.57)$$



Rysunek 5.6: Funkcje aproksymujące zbiór dla zmiennej liczby punktów złamań (programowanie całkowitoliczbowe): $u_1(q_1(x))$ (a), $u_2(q_2(x))$ (b), $v_1(q_1(x))$ (c), $v_2(q_2(x))$ (d)

Funkcje zostały przedstawione również na rysunku 5.6. Widać, że nastąpiło wyraźne uproszczenie wyniku regresji, niestety kosztem wzrostu błędu. Szczególnie w przypadku funkcji cząstkowej dla drugiego kryterium różnica w złożoności jest znaczna. Trudno jest stwierdzić, który sposób daje lepsze wyniki. Z pewnością typowe programowanie liniowe pozwala na znalezienie wyniku dokładnego w dużo krótszym czasie. Z kolei programowanie całkowitoliczbowe wskazuje nam punkty, które w istotny sposób wpływają na funkcje aproksymujące zbiór — w tych miejscach nastąpią złamania. Porównując ostatni wynik z regułami, przede wszystkim umożliwiamy w prosty sposób ciągle odzwierciedlenie zależności monotonicznej. Przy podejściu regułowym, funkcja miała postać

nieciągła, „schodkowa” z powodu tego, że reguła ma wartość decyzji niezależną od odległości, w jakiej dany obiekt się od reguły znajduje.

W celu porównania opisu regułowego z opisem funkcyjnym, przedstawimy dwa wykresy opisujące zależności występujące w znanym już przykładowym zbiorze danych, przedstawione na rysunku 5.7. Na osi poziomej znajduje się wiarygodność pierwszego z kryteriów, na pionowej — drugiego. Zaciemnienie punktów na wykresie przedstawia wiarygodność decyzji. Nauswa się pytanie, co daje nam opis za pomocą funkcji odcinkami liniowych? Z wykresów wynika, że umożliwia lepsze modelowanie profili między klasami, czyli kształtu granicy między kolejnymi wartościami wiarygodności decyzji obiektów. Reguła działa na zasadzie „wszystko albo nic” — wszystkim punktom, które spełniają część warunkową reguły (wspierają ją), przypisują się wartość części decyzyjnej reguły, natomiast na punkty, które części warunkowej nie spełniają, reguła w ogóle nie działa. Funkcje liniowe umożliwiają bardziej płynny wzrost, unikając nagłych skoków wiarygodności decyzji, co wydaje się bliższe rzeczywistości. Przykładowo, gdy reguła mówi: „jeśli $x \geq 0.2$ to $y \geq 0.5$ ”, to obiekt o wartości $x = 0.199$ nie będzie wspierany przez regułę (w konsekwencji przypisana zostanie mu wartość $y = 0$), natomiast obiektowi o wartości $x = 0.201$ zostanie przypisana wiarygodność $y = 0.5$, co oznacza gwałtowny skok preferencji, podczas gdy obiekty są przecież praktycznie nierozróżnialne! W przypadku funkcji liniowej dla małego przyrostu x przyrost y będzie również nieznaczny, gdyż jest liniowo zależny od wartości x , a przede wszystkim *ciągły*.

5.3.5 Konstrukcja reguł na podstawie modelu funkcji odcinkami liniowej

W tym podrozdziale przedstawiony zostanie sposób tworzenia reguł decyzyjnych w oparciu o reprezentację zbioru danych za pomocą funkcji górnego i dolnego przybliżenia odcinkami liniowych. Rozpocznijmy od opisu podstawowego rodzaju reguł, których części decyzyjne są funkcjami liniowymi, nazywanych odąd *regułami liniowymi*. Spróbujmy takie reguły zapisać w postaci implikacji (dla uproszczenia wszystkie kryteria są typu *zysk*):

$$(x_1 \geq \alpha_1 \wedge x_2 \geq \alpha_2 \wedge \dots \wedge x_m \geq \alpha_m) \rightarrow dec(\mathbf{x}) \geq \beta = \sum_{i=1}^m \gamma_i (x_i - \alpha_i) + \gamma_0 \quad (5.58)$$

przy czym $\mathbf{x} = [x_1, \dots, x_m]$ jest punktem w przestrzeni kryteriów (a zarazem przesłanką), $dec(\mathbf{x})$ wartością atrybutu decyzyjnego dla takiego punktu (konkluzją), $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_m]$ punktem umiejscowienia reguły, a $\boldsymbol{\gamma} = [\gamma_0, \gamma_1, \dots, \gamma_m]$ wektorem współczynników kierunkowych związanych z daną regułą.

Zauważmy następnie, że funkcję, będącą sumą funkcji cząstkowych odcinkami liniowych, można przedstawić w postaci zbioru takich reguł. Dokonuje się tego w następujący sposób: w wyniku działania algorytmu opisanego w poprzednim podrozdziale na każdym z kryteriów znajduje się s_j punktów złamań.

Dokonajmy teraz podziału całej przestrzeni na wielowymiarowe prostopadłościany, w taki sposób, że pojedynczy prostopadłościan powstaje z połączenia na każdym z kryteriów dwóch sąsiednich punktów. Przykładowo, dla jednego, j -tego kryterium będą to odcinki o początku w punkcie złamania $h_{k,j}$ i końcu w punkcie $h_{k+1,j}$. Dla dwóch kryteriów, i -tego i j -ego, będą to prostokąty o lewym dolnym narożniku o współrzędnych $[h_{k_i,i}, h_{k_j,j}]$ i prawym górnym o współrzędnych $[h_{k_{i+1,i}}, h_{k_{j+1,j}}]$. Podział każdego z kryteriów na odcinki o początkach i końcach w punktach złamań powoduje podział przestrzeni na siatkę prostopadłościanów. Każdy z punktów takiej siatki jest jednoznacznie przypisany do jednego prostopadłościanu, którego jest początkiem (czyli wierzchołkiem o najmniejszych współrzędnych). W obrębie każdego prostopadłościanu współczynniki funkcji pozostają stałe, zmieniają się jedynie w punktach złamań, czyli gdzieś na siatce. Tym samym, w każdym takim punkcie można postawić regułę liniową, dla której wektor współczynników γ równy jest wektorowi współczynników funkcji w obrębie tego prostopadłościanu. Ścisłej, dla punktu siatki o współrzędnych $[h_{k_1,1}, h_{k_2,2}, \dots, h_{k_m,m}]$ zachodzi dla funkcji $f(\cdot)$:

$$\gamma = [u_0 - \sum_{j=0}^m \sum_{i=0}^{k_j} c_{i,j} h_{i,j}, \sum_{i=0}^{k_1} c_{i,1}, \sum_{i=0}^{k_2} c_{i,2}, \dots, \sum_{i=0}^{k_m} c_{i,m}] \quad (5.59)$$

gdzie dla uproszczenia przyjęto $c_{i,j} = c_{i,j}^+ - c_{i,j}^-$. Dla funkcji $g(\cdot)$ współczynniki reguł będą wyglądały analogicznie, tylko wyrazy $c_{i,j}$ należy zastąpić $d_{i,j}$, a wyraz wolny u_0 wyrazem v_0 . Zwróćmy uwagę, że wartości kolejnych współrzędnych γ_j są wartościami współczynników nachylenia funkcji cząstkowych u_j (lub v_j), a wyraz wolny γ_0 powstaje w wyniku przegrupowania wyrazów stałych w równaniu (5.28) lub (5.29).

Ponieważ reguł liniowych jest dużo, bardzo często będzie się zdarzała sytuacja, w której więcej niż jedna reguła pokrywa nowy obiekt. Zastanówmy się, jak należy rozpatrywać w takim wypadku konkluzje (części decyzyjne). Można pokazać, że jeśli stosujemy reguły liniowe do nowego obiektu i rozważymy wszystkie, które pokrywają taki obiekt, zawsze jest wśród nich jedna, która dominuje wszystkie pozostałe. Istotnie, jest to reguła, która położona jest w punkcie siatki odpowiadającym prostopadłościanowi, w którym leży obiekt. Ścisłej, niech obiekt ma współrzędne $[x_1, \dots, x_m]$ i niech zachodzi:

$$h_{k_j,j} \leq x_j \leq h_{k_{j+1},j} \quad \text{dla wszystkich } j \quad (5.60)$$

gdzie $h_{k_j,j}$ i $h_{k_{j+1},j}$ to najbliższy na lewo i najbliższy na prawo punkt złamania na j -tym kryterium. Reguły, które pokrywają taki obiekt, muszą być przez ten obiekt dominowane, stąd współrzędne punktu umiejscowienia takiej reguły $[\alpha_1, \dots, \alpha_m]$ muszą spełniać zależność:

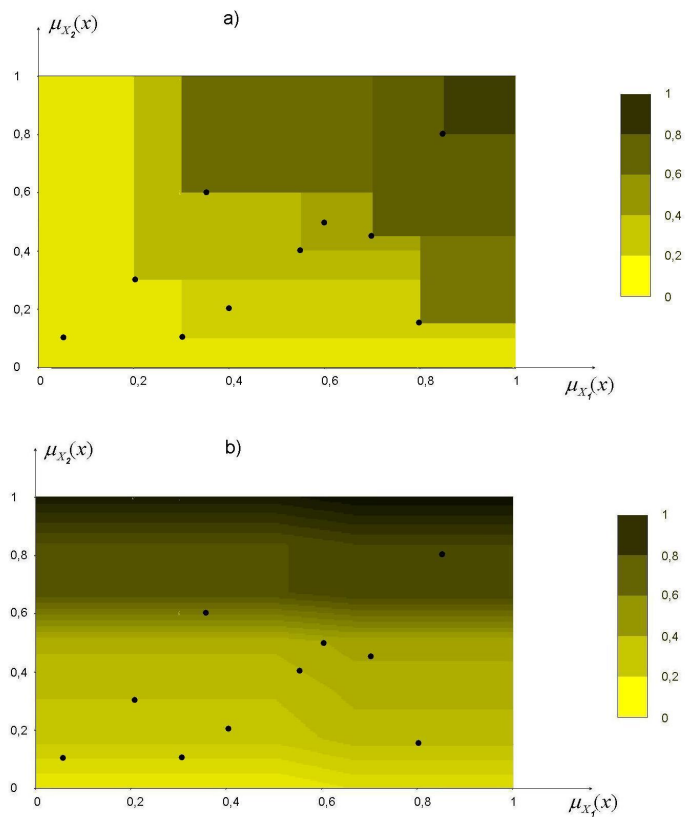
$$\alpha_j \leq x_j \quad \text{dla wszystkich } j \quad (5.61)$$

Dodatkowo, każda ze współrzędnych α_j może przyjmować tylko wartości $h_{i,j}$ gdyż reguła musi leżeć na siatce wyznaczonej przez punkty złamań. Biorąc pod uwagę powyższe stwierdzenie oraz ograniczenia (5.60) i (5.61), zachodzi:

$$\alpha_j \leq h_{k_j, j} \quad \text{dla wszystkich } j \quad (5.62)$$

W szczególności reguła o punkcie umiejscowienia $\alpha = [h_{k_1, 1}, \dots, h_{k_m, m}]$ zgodnie z (5.62) dominuje wszystkie pozostałe. Wartością decyzji dla nowego obiektu będzie wartość wyznaczona za pomocą tej właśnie reguły.

Reguły przedstawione poniżej pozwalają bezpośrednio wskazać lokalny, liniowy trend funkcji w okolicach danego punktu. Podstawowym problemem jest szybko rosnąca liczba reguł, równa liczbie sposobów na jakie można wybrać punkty na każdym z kryteriów, czyli $s_1 \cdot s_2 \cdot \dots \cdot s_n$. Liczba ta rośnie wykładniczo wraz z liczbą kryteriów i wielomianowo wraz z liczbą punktów złamań. Ponownie, dużą pomocą wydaje się tutaj dokonanie uprzedniej redukcji kryteriów.



Rysunek 5.7: Porównanie dwóch modeli zbioru: a) za pomocą regresji liniowej
b) za pomocą indukcji reguł

Rozdział 6

Podsumowanie

Tematem przewodnim pracy są reguły gradualne wyrażające ogólną zależność „im bardziej/mniej x jest X , tym bardziej x jest Y ”. Z uwagi na to, że powyższe stwierdzenie ma bardzo szeroki zakres znaczeniowy, można postać reguł uściślić. Dokonano tego korzystając przede wszystkim z gotowego już podejścia opisanego w pracach S. Greco, M. Inuiguchi’ego i R. Słowińskiego [14, 15, 13], przedstawiając podstawowe idee w rozdziale 3, po uprzednim zapoznaniu się z niezbędnymi pojęciami dotyczącymi logiki rozmytej, teorii zbiorów przybliżonych oraz reguł decyzyjnych. We wspomnianym rozdziale przedstawiona została bardziej formalna i silniejsza definicja reguł gradualnych oraz rozmyte przybliżone podejście będące podstawą indukcji reguł. Podejście to łączy zastosowanie zbiorów rozmytych z przybliżonymi, dzięki czemu pozwala na radzenie sobie z danymi zawierającymi wartości niespójne, niepewne lub niedokładne.

To właśnie indukowanie funkcji $f(\cdot)$ i $g(\cdot)$ poprzez zastosowanie przybliżonego rozmytego podejścia było podstawą dla wszystkich kolejnych rozdziałów i wszystkich pomysłów rozwinięcia metod opartych o reguły gradualne, jakie się w pracy pojawiły. W późniejszych rozdziałach korzystano również z ogólnej formy zapisu reguł, gradualnych, oraz z dwóch postawowych pojęć, zdefiniowanych w rozdziale 3: *poprawności* oraz *ostrożności*. Pierwsze z nich pozwala ustalić pewne wymagania, które powinna spełniać każda reguła gradualna, opisująca dany zbiór w sposób spójny, drugie, opisujące reguły dopuszczalne (poprawne), stopniuje je wprowadzając skalę względem ostrożności wyciąganych wniosków. W tym samym rozdziale pojawia się zasada, zgodnie z którą jesteśmy w stanie wybrać jedną, konkretną funkcję gradualną do opisu danych: będzie to funkcja najbardziej ostrożna.

Oczywiście, funkcja wyciągająca możliwie najostrożniejsze wnioski z pewnością ułatwia poprawne klasyfikowanie późniejszych obiektów, ale szacowanie na nich dokonane jest słabsze i mniej dokładne. Drugi problem to wymagana przejrzystość i mały stopień komplikacji opisu zbioru — maksymalnie ostrożne reguły zwykle nie spełniają takich warunków. Poprzez analogię do podejścia opartego o relację dominacji (DRSA) można próbować, opierając się na funkcjach $f(\cdot)$ i $g(\cdot)$, indukować reguły w taki sposób, aby zachowały poprawność,

a zarazem wyciągały już mniej ostrożne wnioski. Podstawowym celem było tu zwrócenie uwagi na fakt, że mimo stosowania algorytmów redukcji atrybutów lub indukcji reguł używanych w podejściu DRSA, czy też definiowanie heurystycznych miar oceniających reguły, pozwalających na ich selekcje (czyli metod pozornie nie związanych z podejściem opartym o reguły gradualne) opis zbioru pozostał dokładnie ten sam — nadal za pomocą funkcji gradualnych dolnego i górnego przybliżenia, nadal funkcji poprawnych, ale już nie najbardziej ostrożnych. Tym samym opisywane podejście pozostało spójne z zaproponowanym przez S. Greco, M. Inuiguchi’ego i R. Słowińskiego.

Zastosowano tutaj dwie metody tworzenia mniej ostrożnego opisu zbioru danych — poprzez redukcję atrybutów (która z kolei opierała się na pojęciu reduktu) oraz przez indukcję reguł za pomocą algorytmu ADRIA, bazującego na macierzy dominacji. Drugi ze sposobów wydawał się bardziej ogólny i skuteczny, jednak w wyniku działania dawał wyczerpujący zbiór reguł, a przez to nadmiarowy. Stąd zaproponowano dodatkowo pewne miary pozwalające na wybór zbioru reguł w taki sposób, by uniknąć redundacji w zbiorze. Miary skonstruowane zostały w oparciu o wartości dotyczące jednej reguły (związane z pokrywaniem przykładami) oraz dotyczące par reguł (związane z szacowaniem podobieństwa).

Niestety liczba reguł nadal była dość spora z powodu zbyt gęstego podziału na klasy, co w naturalny sposób komplikowało i zwiększało opis zbioru. Ponieważ wszelkie wartości pojawiające się w opisie wiarygodności na wszystkich kryteriach i decyzji, są numeryczne, w rozdziale 5 pojawił się pomysł, aby fakt ten wykorzystać i wprowadzić możliwość porównywania odległości między obiektami na kryteriach i wiarygodności decyzji. Było to założenie odbiegające od idei towarzyszących dotychczas podejściu opartemu na regułach gradualnych, w ramach którego korzysta się wyłącznie z zależności porządkowych. Próbą rekompensowania przyjęcia mocniejszych warunków może być za to uproszczenie opisu zbioru poprzez utożsamianie podobnych wartości (odpowiadające koncepcyjnie dyskretyzacji) oraz zupełnie nowy sposób zapisu zależności w zbiorze za pomocą funkcji odcinkami liniowych. Oba te podejście nadal wiążą się z regułami gradualnymi, chociażby dlatego, że pierwsze z nich jest tylko innymi sposobem selekcji takich reguł, a w przypadku drugim tworzymy funkcje odcinkami liniowe, monotoniczne, które można utożsamić z funkcjami gradualnymi. Jeśli funkcje takie uda się wyznaczyć bez błędu aproksymacji, będzie to kolejny rodzaj reguł gradualnych, poprawnych ale nie maksymalnie ostrożnych, czyli spójnych z podejściem z rozdziału 3. Jedyna różnica, jaka się pojawia to fakt, że funkcje odcinkami liniowe tworzone są na podstawie porównywania odległości między obiektami, stąd bez używania skali, przy której możemy się o takich odległościach wypowiadać, szacowania takich funkcji wydają się niczym nieuzasadnione. Tym samym jesteśmy zmuszeni do przyjęcia skali ilorazowej.

Innym wynikiem, uzyskanym w ramach pracy nad powyższymi zagadnieniami, jest system indukcji reguł gradualnych, pozwalający na zastosowanie podstawowych idei w przypadkach praktycznych. Został on opisany w dodatku do niniejszej pracy.

Myślę, że metodologia związana z regułami gradualnymi, jest oryginalnym, a

zarazem bardzo ogólnym sposobem opisu zbioru danych, szczególnie dla decyzji w postaci zmiennej ciągłej. W przyszłości praca nad nią będzie z pewnością kontynuowana, o czym świadczy wiele możliwości potencjalnego jej rozwinięcia.

Bibliografia

- [1] J. Błazewicz, W. Cellary, R. Słowiński, J. Węglarz, *Badania operacyjne dla informatyków*. WNT, Warszawa 1983.
- [2] S. Brandt, *Metody statystyczne i obliczeniowe analizy danych*. PWN, Warszawa 1976.
- [3] P. Cichosz, *Systemy uczące się*. WNT, Warszawa 2000.
- [4] G. Dantzig. *Linear programming and extensions*. Princeton University Press, Princeton NJ, 1963.
- [5] K. Dembczyński, *Algorytmy generowania reduktów i reguł decyzyjnych przy hierarchicznej strukturze atrybutów i kryteriów*. Praca Magisterska, Instytut Informatyki, Poznań 2001.
- [6] K. Dembczyński, R. Pindur, R. Susmaga, *Generation of Exhaustive Set of Rules with Dominance-based Rough Set Approach*. Elsevier Science B. V., 2003.
- [7] D. Dubois, H. Prade, *What are fuzzy rules and how to use them*. *Fuzzy Sets and Systems* 84 (1996), 169-185
- [8] D. Dubois, H. T. Nguyen, H. Prade, M. Sugeno, *The real contribution of fuzzy systems*. Hung T. Nguyen, Michio Sugeno (eds.), *Fuzzy Systems: Modeling and Control*. The Handbook of Fuzzy Sets Series (2), 1998, 1-17
- [9] D. Dubois, L. Ughetto, H. Prade, *Implicative and conjunctive fuzzy rules — A tool for reasoning from knowledge and examples*. *Proceedings of the 16th American National Conference on Artificial Intelligence (AAAI'99)*, 214-219, Orlando, Florida, USA, Juillet 1999.
- [10] General Algebraic Modeling System (GAMS), <http://www.gams.com>
- [11] *GOAL Project: Otwarta platforma programowa dla systemów eksploracji danych*, <http://www-idss.cs.put.poznan.pl/~GoalProject>
- [12] S. I. Gass, *Programowanie liniowe*. PWN, Warszawa 1980.

- [13] S. Greco, M. Inuiguchi, R. Słowiński, *Fuzzy rough sets, gradual decision rules and approximate reasoning*. Manuskrypt, 2004.
- [14] S. Greco, M. Inuiguchi, R. Słowiński, *A new proposal for fuzzy rough approximations and gradual decision rule representation*. W: D. Dubois, J. Grzymala-Busse, M. Inuiguchi, L. Polkowski (eds.), *Rough Fuzzy and Fuzzy Rough Sets: Fuzzy and Rough and Fuzzy along Rough*. Lecture Notes in Computer Science, vol. 3135, Springer-Verlag, Berlin, 2004, pp. 332-355
- [15] S. Greco, M. Inuiguchi, R. Słowiński, *Fuzzy rough sets and multiple-premise gradual decision rules*. *International Workshop on Fuzzy Logic and Applications (WILF 2003)*, Napoli, Włochy, 9 -11 października 2003, pp. 56-61
- [16] S. Greco, B. Matarazzo, R. Słowiński, *Rough approximation of a preference relation by dominance relations*. *European Journal of Operational Research*, 117 (1999), 63-83.
- [17] S. Greco, B. Matarazzo, R. Słowiński, *Extension of the rough set approach to multicriteria decision support*. *INFOR* 38, 2000, 161-196.
- [18] S. Greco, B. Matarazzo, R. Słowiński, *Rough sets theory for multicriteria decision analysis*. *European Journal of Operational Research* 129 (2001), 1-47.
- [19] S. Greco, B. Matarazzo, R. Słowiński, *Induction of decision rules for explanation and support of multicriteria choice and ranking*. *Proc. 2nd Int. Conference on Decision Support for Telecommunications and Information Society (DSTIS 2002)*, Warszawa, 5-7 września 2002, 51-61.
- [20] J. Hopcroft, J. Ullman, *Wprowadzenie do teorii automatów, języków i obliczeń*. PWN, 2003
- [21] E. Jacquet-Lagrange, J. Siskos, *Assessing a Set of Additive Utility Functions for Multi-criteria Decision-making, the UTA Method*, *European Journal of Operational Research*, 10(2), 1982, 151-164.
- [22] V. Kreinovich, H. Nguyen, *Methodology of fuzzy control: an introduction*. W: Hung T. Nguyen, Michio Sugeno (eds.), *Fuzzy Systems: Modeling and Control*. The Handbook of Fuzzy Sets Series (2), 1998.
- [23] R. S. Michalski, *O naturze uczenia się - problemy i kierunki badawcze*. *Informatyka*, 2, 1998.
- [24] *lp-solve* (Michiel Berkelaar), ftp://ftp.ics.ele.tue.nl/pub/lp_solve/
- [25] Z. Pawlak, *Rough sets*. *International Journal of Computer and Information Sciences*, 11 (1982), 341-356.
- [26] Z. Pawlak, *Rough set Elements*. W: L. Polkowski, A. Skowron (eds.) *Rough Sets in Knowledge Discovery Heidelberg*. Physica-Verlag, 1998, 10-30.

- [27] B. Roy, *Wielokryterialne wspomaganie decyzji*. WNT, Warszawa 1990.
- [28] M. Stark, A. Mostowski, *Algebra liniowa*, PWN, Warszawa 1977.
- [29] A. Skowron, C. Rauszer, *The discernibility matrices and functions in information systems*. W: Słowiński R. (red.), *Intelligent Decision Support. Handbook of Applications and Advances of the Rough Set Theory*, Kluwer Academic Publishers, Dordrecht (1992), 331-362.
- [30] R. Słowiński, *Rozszerzenie teorii zbiorów przybliżonych na atrybuty ze skalą preferencji*. Raporty Politechniki Poznańskiej, RB-014/2000, Poznań 2000.
- [31] J. Stefanowski, *Algorytmy indukcji reguł decyzyjnych w odkrywaniu wiedzy*. Rozprawa habilitacyjna, Wydawnictwo Politechniki Poznańskiej, Seria Rozprawy 361, Poznań (2001).
- [32] J. Stefanowski, *On Rough Set Based Approaches to Induction of Decision Rules*. W: L. Polkowski, A. Skowron (eds.) *Rough Sets in Knowledge Discovery Heidelberg*. Physica-Verlag, 1998, 500-529.
- [33] R. Susmaga, *Experiments in Incremental Computation of Reducts*. W: L. Polkowski, A. Skowron (eds.) *Rough Sets in Knowledge Discovery Heidelberg*. Physica-Verlag, 1998, 530-553.
- [34] R. Susmaga, *Algorytmy dokładnej redukcji danych oparte na teorii zbiorów przybliżonych*. Rozprawa doktorska, Politechnika Poznańska, 2001.
- [35] K. Tanaka, M. Sugeno, *Introduction to fuzzy modeling*. W: Hung T. Nguyen, Michio Sugeno (eds.), *Fuzzy Systems: Modeling and Control*. The Handbook of Fuzzy Sets Series (2), 1998, 63-89.
- [36] R. Yager, D. Filev, *Podstawy modelowania i sterowania rozmytego*. WNT, 1995
- [37] R. Yager, D. Filev, *Fuzzy rule based models and approximate reasoning*. W: Hung T. Nguyen, Michio Sugeno (eds.), *Fuzzy Systems: Modeling and Control*. The Handbook of Fuzzy Sets Series (2), 1998, 90-133.
- [38] L. A. Zadeh, *Fuzzy sets. Information and Control*, nr 8 (1965), 1965.

Dodatek: Opis programu i instrukcja użytkownika

6.1 Opis ogólny

Jako dodatek do niniejszej pracy powstał program *Gradual*, implementujący opisane algorytmy. Jego funkcjonalność obejmuje cały zakres podejść zawartych w pracy, począwszy od wyznaczania wiarygodności dolnych i górnych przybliżeń, skończywszy na regresji liniowej.

System został napisany w języku *Java*, co umożliwia uruchomienie go w ramach dowolnego systemu operacyjnego i na dowolnej platformie sprzętowej. Mimo, że wykorzystuje zewnętrzne biblioteki (algorytmy zawarte w *GOAL-Project* oraz pakiet funkcji programowania liniowego), został zaprojektowany i napisany jako niezależna całość, co pozwala go uruchomić bez konieczności posiadania dodatkowych narzędzi, za wyjątkiem maszyny wirtualnej Javy. Pełna dokumentacja techniczna, dokładnie opisująca kod, została dołączona jako dodatek na płycie CD wraz z systemem. Znajduje się tam również dokumentacja użytkownika, w sposób wyczerpujący opisująca instalację, sposób uruchomienia oraz wszystkie funkcje programu. Tutaj skupimy się na opisie funkcjonalności systemu i ogólnym przeglądzie jego możliwości.

Podstawą działania wszystkich algorytmów wchodzących w skład systemu jest podejście oparte na regułach gradualnych. Mając na wejściu dane w postaci tablicy decyzyjnej, tzn. zbiór obiektów wraz z wartościami na atrybutach i wiarygodnością decyzji dla każdego obiektu, można uzyskać następujące wyniki:

- edytować graficznie tablicę decyzyjną i w przejrzysty sposób wyznaczać wiarygodność decyzji,
- skonstruować na każdym z atrybutów pojęcie (zmienną lingwistyczną), będące zbiorem rozmytym, wraz z funkcją przynależności (wiarygodności) dla tego pojęcia,
- przekształcić wartości na atrybutach na wiarygodności przynależności do zbiorów rozmytych, mające charakter kryteriów,
- wyznaczyć wiarygodność przynależności do górnego i dolnego przybliżenia dla każdego obiektu,



Rysunek 6.1: System indukcji reguł gradualnych *Gradual*.

- generować reguły decyzyjne na podstawie przynależności do dolnych i górnych przybliżeń,
- generować wyczerpujący zbiór reguł minimalnych za pomocą algorytmu indukcji,
- dokonać selekcji zbioru reguł minimalnych za pomocą jednego z trzech algorytmów selekcji,
- skonstruować opis zbioru poprzez regresję funkcjami odcinkami liniowymi,
- przeglądać wygenerowane zbiory reguł wraz z obiektami wspierającymi wybrane reguły,
- prezentować graficznie wszystkie opisy zbioru (za pomocą reguł i regresji) na przestrzeni kryteriów, poprzez projekcję na podprzestrzeń jedno- lub dwuwymiarową,
- klasyfikować nowe przykłady.

Wszystkie wymienione funkcje zostaną w dalszej części omówione bardziej szczegółowo.

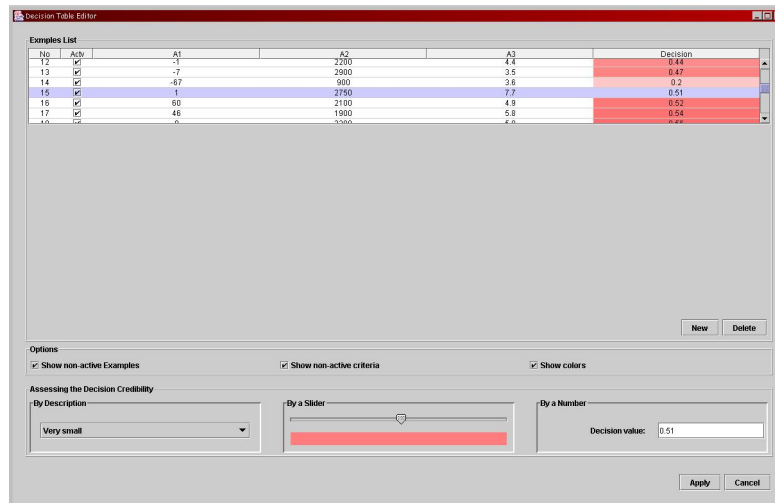
Celem uproszczenia pracy użytkownikom systemu, został opracowany format pliku, przyjmowany na wejściu przez system. Plik zawiera pełne dane dotyczące tablicy decyzyjnej, opisu atrybutów, zmiennych lingwistycznych oraz ich funkcji przynależności. Format pliku jest bardzo prosty, gdyż jest formatem tymczasowym — zostanie w późniejszym czasie zastąpiony standardowym formatem przeznaczonym do tego typu narzędzi (ISF), stąd wymyślanie skomplikowanego formatowania, niezgodnego ze standardami, miało się z celem.

6.2 Funkcje programu

Zostaną omówione wszystkie podstawowe funkcje oferowane przez system.

6.2.1 Edycja tablicy decyzyjnej

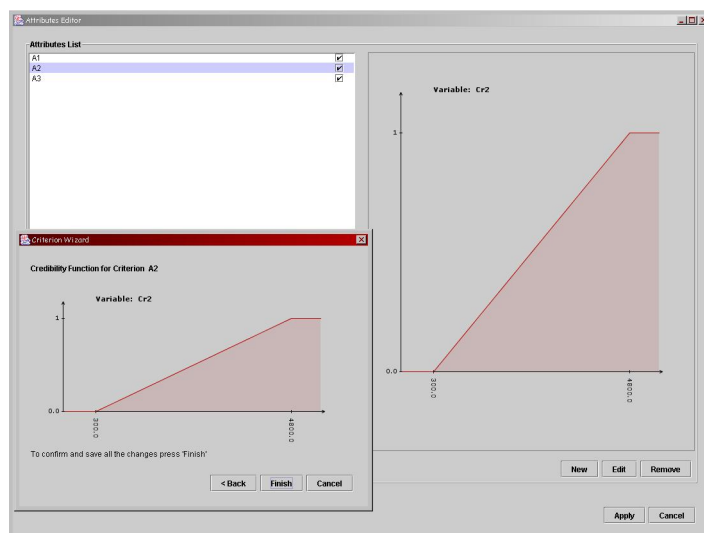
Edytor pozwala na wnoszenie zmian w tablicy decyzyjnej. W tablicy zostają wyświetlone obiekty wraz z wartościami na atrybutach oraz wiarygodnością decyzji (dodatkowo podświetloną kolorem, którego nasycenie odpowiada wartości wiarygodności). Każde z pól można swobodnie edytować, a dodatkowo wartości decyzji, przez wzgląd na fakt, że mogą być silnie subiektywne, mogą zostać wyznaczone również dwoma innymi metodami: poprzez ustawienie suwaka określającego intensywność preferencji lub poprzez wybór jednego z określeń słownych (np. „niski”, „wysoki”, „pewny”, itp.), do których przyporządkowane są odpowiednie liczby.



Rysunek 6.2: Edytor tablicy decyzyjnej.

6.2.2 Konstrukcja kryteriów

Kolejny edytor, pozwalający na wprowadzanie zmian związanych z atrybutami i zbiorami rozmytymi. Istnieje możliwość dodawania, usuwania lub edytowania atrybutów. Z każdym atrybutem należy związać pewien zbiór rozmyty (zmienną lingwistyczną), do którego przynależność ma charakter kryterium. Do modelowania funkcji przynależności służy specjalny *wizard*, pozwalający, poprzez zadawanie prostych pytań (np. „dla jakiej wartości można być pewnym, że spełnia ona konstruowane pojęcie?”), na utworzenie trzech rodzajów funkcji: malejącej, rosnącej oraz najbardziej skomplikowanej sześciopunktowej.



Rysunek 6.3: Edytor Atrybutów i kryteriów.

6.2.3 Przekształcanie atrybutów, dolne i górne przybliżenia

Mając daną tablicę decyzyjną oraz zbiór atrybutów wraz ze zbiorami rozmytymi i funkcjami przynależności, można przedstawić zbiór obiektów w przestrzeni wartości przynależności do zmiennych lingwistycznych, zamiast, jak dotychczas, w przestrzeni wartości na atrybutach. Wspomniane przynależności mają charakter kryteriów, co pozwala z kolei na wyznaczenie wiarygodności przynależności do górnych i dolnych przybliżeń.

6.2.4 Generowanie i przeglądanie reguł decyzyjnych

W celu znalezienia opis zbioru danych, generuje się gradualne reguły decyzyjne. W systemie możliwa jest generacja trzech zbiorów takich reguł. Pierwszy zawiera reguły, opisujące funkcje gradualne maksymalnie ostrożne, powstałe bez stosowania indukcji. Drugi zbiór zawiera reguły wygenerowane algorytmem indukcji i jest to wyczerpujący zbiór reguł minimalnych. Trzeci z kolei to podzbiór zbioru wyczerpującego, zawierający reguły będące wynikiem zastosowania jednego z trzech rodzajów selekcji: na podstawie wsparcia, na podstawie stopnia przekrywania oraz na podstawie miary odległości. Trzeci rodzaj selekcji wymaga podania dodatkowo dwóch wartości progowych, określających krok dyskretyzacji w przestrzeni kryteriów i wiarygodności decyzji. Istnieje jeszcze jeden parametr — warunek stopu, pozwalający (dla każdego rodzaju selekcji) na do pewnego stopnia niespójny opis zbioru (parametr jest procentową ilością obiektów, z jakimi opis regułowy powinien być spójny).

No	Cr1	Cr2	Cr3	Decision	f00: Lower Approx.	g00: Upper Approx.
1	0	0	0	0	0	0
2	0.194	0	0.093	0.04	0.04	0.04
3	0.056	0.1	0.328	0.1	0.1	0.2
4	0.011	0.133	0.012	0.12	0.12	0.12
5	0.111	0	0.593	0.15	0.15	0.15
6	0.033	0.071	0	0.2	0.1	0.2
7	0.394	0.033	0.093	0.2	0.2	0.2
8	0.178	0.422	0.198	0.24	0.24	0.24
9	0.567	0.244	0.244	0.32	0.32	0.32
10	0.420	0.822	0.233	0.39	0.39	0.39
11	0.772	0.487	0.314	0.41	0.41	0.49
12	0.494	0.422	0.372	0.44	0.44	0.49
13	0.461	0.578	0.207	0.47	0.47	0.47
14	0.128	0.133	0.279	0.48	0.41	0.48
15	0.586	0.544	0.758	0.51	0.51	0.67
16	0.933	0.4	0.42	0.52	0.52	0.57
17	0.756	0.356	0.535	0.54	0.54	0.57
18	0.544	0.844	0.547	0.56	0.56	0.56
19	0.728	0.233	0.328	0.57	0.57	0.57
20	0.789	0.578	0.888	0.6	0.6	0.69
21	0.556	0.822	0.5	0.63	0.63	0.63
22	0.283	0.438	0.651	0.67	0.67	0.67
23	0.75	0.523	0.512	0.69	0.69	0.69
24	0.383	0.798	0.674	0.72	0.72	0.72
25	0.981	0.9	0.42	0.74	0.74	0.74
26	0.594	0.867	0.36	0.79	0.79	0.79
27	0.828	0.624	0.547	0.8	0.8	0.8
28	0.889	0.822	0.651	0.83	0.83	0.88
29	0.828	0.722	0.802	0.86	0.86	0.86
30	0.878	0.822	0.57	0.88	0.83	0.88
31	0.65	0.978	0.884	0.93	0.93	0.93
32	1	0.844	1	0.96	0.96	0.96
33	1	1	1	1	1	1

Rysunek 6.4: Wartości przynależności do zbiorów rozmytych (kryteria) oraz wiarygodność przynależności do dolnego i górnego przybliżenia.

Po wygenerowaniu reguł (za pomocą zewnętrznych bibliotek pochodzących z systemu *GOAL-Project*), można obejrzeć ich listę dla każdego ze zbiorów. Podświetlenie danej reguły umożliwia podgląd obiektów, które są przez taką regułę pokryte.

Type of Rules: Lower and upper approximation rules

Rules generated from dominance matrix:

- Cr1 == 0.428 & Cr2 == 0.822 & Cr3 == 0.233 -> dec == 0.38
- Cr1 == 0.772 & Cr2 == 0.467 & Cr3 == 0.314 -> dec == 0.41
- Cr1 == 0.494 & Cr2 == 0.422 & Cr3 == 0.372 -> dec == 0.44
- Cr1 == 0.461 & Cr2 == 0.578 & Cr3 == 0.267 -> dec == 0.47
- Cr1 == 0.128 & Cr2 == 0.133 & Cr3 == 0.278 -> dec == 0.41
- Cr1 == 0.586 & Cr2 == 0.544 & Cr3 == 0.758 -> dec == 0.51
- Cr1 == 0.833 & Cr2 == 0.4 & Cr3 == 0.42 -> dec == 0.52
- Cr1 == 0.756 & Cr2 == 0.356 & Cr3 == 0.535 -> dec == 0.54
- Cr1 == 0.544 & Cr2 == 0.844 & Cr3 == 0.547 -> dec == 0.56
- Cr1 == 0.728 & Cr2 == 0.233 & Cr3 == 0.328 -> dec == 0.57
- Cr1 == 0.789 & Cr2 == 0.578 & Cr3 == 0.888 -> dec == 0.6
- Cr1 == 0.556 & Cr2 == 0.822 & Cr3 == 0.5 -> dec == 0.63
- Cr1 == 0.283 & Cr2 == 0.438 & Cr3 == 0.651 -> dec == 0.67
- Cr1 == 0.75 & Cr2 == 0.523 & Cr3 == 0.512 -> dec == 0.69
- Cr1 == 0.383 & Cr2 == 0.798 & Cr3 == 0.674 -> dec == 0.72
- Cr1 == 0.981 & Cr2 == 0.9 & Cr3 == 0.42 -> dec == 0.74
- Cr1 == 0.594 & Cr2 == 0.867 & Cr3 == 0.36 -> dec == 0.79
- Cr1 == 0.828 & Cr2 == 0.624 & Cr3 == 0.547 -> dec == 0.8
- Cr1 == 0.889 & Cr2 == 0.822 & Cr3 == 0.651 -> dec == 0.83
- Cr1 == 0.828 & Cr2 == 0.722 & Cr3 == 0.802 -> dec == 0.86
- Cr1 == 0.878 & Cr2 == 0.822 & Cr3 == 0.57 -> dec == 0.88
- Cr1 == 0.65 & Cr2 == 0.978 & Cr3 == 0.884 -> dec == 0.93
- Cr1 == 1 & Cr2 == 0.844 & Cr3 == 1 -> dec == 0.96
- Cr1 == 1 & Cr2 == 1 & Cr3 == 1 -> dec == 1

Examples covered by selected rule:

No	Cr1	Cr2	Cr3	Decision	f00: Lower Approx.	g00: Upper Approx.
1	0.428	0.822	0.233	0.38	0.38	0.38
2	0.544	0.844	0.547	0.56	0.56	0.56
3	0.556	0.822	0.5	0.63	0.63	0.63
4	0.981	0.9	0.42	0.74	0.74	0.74
5	0.594	0.867	0.36	0.79	0.79	0.79
6	0.828	0.624	0.547	0.8	0.8	0.8
7	0.889	0.822	0.651	0.83	0.83	0.88
8	0.828	0.722	0.802	0.86	0.86	0.86
9	0.878	0.822	0.57	0.88	0.83	0.88
10	0.65	0.978	0.884	0.93	0.93	0.93
11	1	0.844	1	0.96	0.96	0.96
12	1	1	1	1	1	1

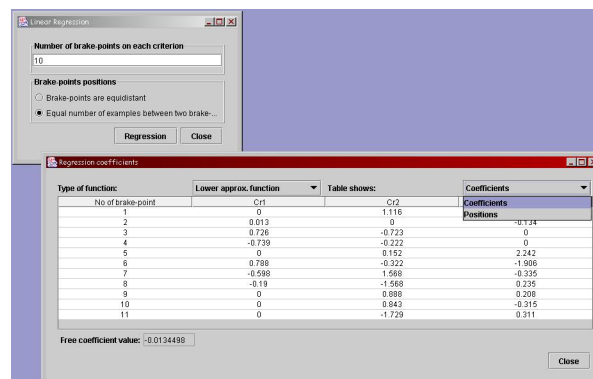
Rysunek 6.5: Wygenerowane reguły decyzyjne.

6.2.5 Regresja

System umożliwia również skonstruowanie opisu zbioru obiektów za pomocą funkcji będącej sumą funkcji cząstkowych odcinkami liniowymi. Jednym z pa-

rametrów, które należy ustalić, jest liczba punktów charakterystycznych takich funkcji, czyli liczba punktów, w którym następuje złamanie. Inną kwestią jest samo rozmieszczenie takich punktów — mogą one zostać ustawione na osi kryteriów tak, aby między każdymi kolejnymi dwoma punktami odległość była stała (co oznacza podział osi kryterium na równe odcinki). Takie punkty są *równoodległe*. Inny sposób to utworzenie listy wszystkich wartości, jakie przyjmują obiekty na danym kryterium i ustawienie punktów w ten sposób, aby między każdymi kolejnymi punktami liczba wartości była stała. Przedziały utworzone przez takie punkty są *równoliczne*.

Regresja jest realizowana za pomocą bibliotek zewnętrznych *lp-solve*. Po jej zakończeniu istnieje możliwość przejrzania wyników: rozmieszczenia wszystkich punktów charakterystycznych oraz względnym zmian współczynników kierunkowych w tych punktach dla każdego z kryteriów.



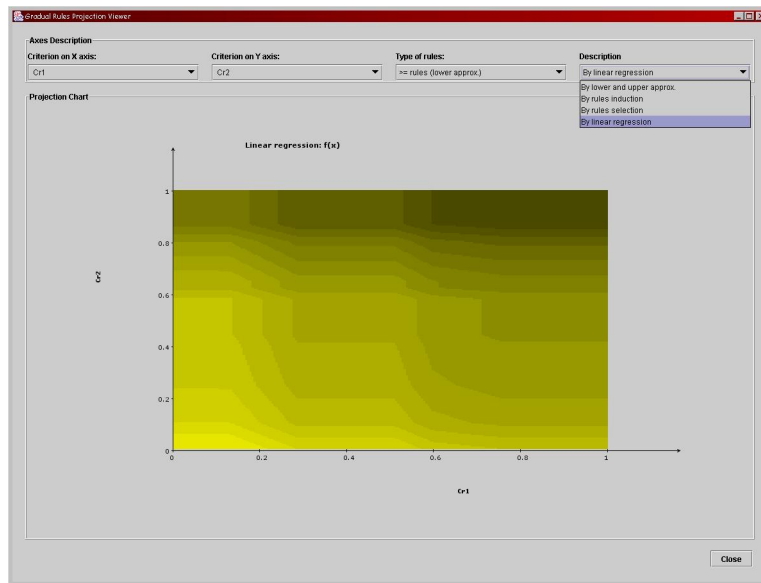
Rysunek 6.6: Regresja liniowa.

6.2.6 Prezentacja graficzna wyników

Istnieje możliwość graficznej prezentacji opisu zbioru. Dokonuje się tego poprzez redukcję przestrzeni kryteriów do podprzestrzeni jedno- lub dwuwymiarowej. W pierwszym przypadku, przy redukcji do osi jednego z kryteriów, X_i , można utworzyć wykres zależności wiarygodności decyzji od wartości na kryterium, $\mu_Y(x) = f(\mu_{X_i}(x))$. W drugim przypadku, przy redukcji do płaszczyzny (dwóch kryteriów), X_i i X_j , można utworzyć wykres zależności wiarygodności decyzji od wartości na obu kryteriach, $\mu_Y(x) = f(\mu_{X_i}(x), \mu_{X_j}(x))$, przy czym wykres jest przedstawiony na płaszczyźnie kryteriów, a wiarygodność decyzji oznaczana jest kolorami — im kolor ciemniejszy, tym wiarygodność większa.

6.2.7 Klasyfikacja

Po konstrukcji opisu zbioru istnieje możliwość klasyfikacji nowych przykładów, o nieznanach wartościach wiarygodności decyzji. Poprzez wpisanie warto-



Rysunek 6.7: Graficzna prezentacja opisu zbioru.

ści, jakie nowy obiekt przyjmuje na atrybutach oraz wybrania opisu, w ramach którego odbędzie się klasyfikacja (trzy rodzaje reguł lub regresja), otrzymujemy górny i dolny koniec przedziału (poprzez zastosowanie odpowiednio funkcji górnego i dolnego przybliżenia), w obrębie którego powinna znaleźć się wartość wiarygodności decyzji nowego przykładu.