

Autoreferat

1 Imię i Nazwisko: Beata Zielosko

Instytut Informatyki, Uniwersytet Śląski
ul. Będzińska 39, 41-200 Sosnowiec

2 Posiadane dyplomy

- **2000: tytuł licencjata informatyki** (specjalność: bazy danych i systemy inteligentne), Wydział Techniki, Uniwersytet Śląski;
- **2002: tytuł magistra informatyki** (specjalność: bazy danych i systemy inteligentne), Wydział Techniki, Uniwersytet Śląski;
- **2008: tytuł doktora nauk technicznych w zakresie informatyki**, Wydział Informatyki i Nauki o Materiałach (przed zmianą nazwy Wydział Techniki), Uniwersytet Śląski.

Rozprawa doktorska: *Generowanie i optymalizacja częściowych reguł decyzyjnych* (ang. *Construction and Optimization of Partial Decision Rules*) napisana pod kierunkiem prof. dr hab. Mikhaila Moshkova. Rozprawa uzyskała wyróżnienie oraz Nagrodę Naukową Rektora Uniwersytetu Śląskiego w Katowicach dla autorów najlepszych prac doktorskich.

3 Zatrudnienie w jednostkach naukowych

- **01.03.2009 - teraz** – adiunkt, Instytut Informatyki, Wydział Informatyki i Nauki o Materiałach, Uniwersytet Śląski;
- **08.02.2011 - 28.03.2013** – starszy naukowiec, Computer, Electrical and Mathematical Sciences & Engineering Division, King Abdullah University of Science and Technology, Arabia Saudyjska;
- **01.10.2002 - 28.02.2009** – asystent, Instytut Informatyki, Wydział Informatyki i Nauki o Materiałach (przed zmianą nazwy Wydział Techniki), Uniwersytet Śląski;
- **01.10.2005 - 30.03.2010 oraz 01.10.2014 - 28.02.2015** – wykładowca w Wyższej Szkole Technologii Informatycznych w Katowicach;
- **08.10.2001 - 30.09.2002** – st. technik, Instytut Informatyki, Wydział Techniki, Uniwersytet Śląski.

Spis treści

4	Podstawowe osiągnięcie	3
4.1	Wprowadzenie	4
5	Porównanie reguł decyzyjnych, drzew i testów	6
5.1	Powiązania pomiędzy regułami decyzyjnymi, drzewami i testami	7
5.2	Granice złożoności reguł decyzyjnych, drzew i testów	10
5.3	Algorytmy dla konstruowania reguł decyzyjnych	12
5.4	Tablice decyzyjne z wielowartościową decyzją	15
5.5	Przybliżone reguły decyzyjne, drzewa i testy	18
5.6	Uczenie nadzorowane – klasyfikatory oparte na regułach decyzyjnych	22
5.7	Lokalne podejście dla badania drzew decyzyjnych i systemów reguł	25
5.8	Drzewa decyzyjne i reguły nad liniowymi systemami informacyjnymi	31
6	Rozszerzenia podejścia opartego na dynamicznym programowaniu dla analizy i optymalizacji reguł decyzyjnych	32
6.1	Podejście oparte na dynamicznym programowaniu dla sekwencyjnej optymalizacji reguł decyzyjnych	32
6.2	Zmodyfikowany algorytm dynamicznego programowania dla optymalizacji przybliżonych reguł decyzyjnych względem pokrycia	39
6.3	Powiązania pomiędzy długością i pokryciem reguł decyzyjnych ..	42
7	Podejście zachłanne i podejście oparte na dynamicznym programowaniu dla optymalizacji reguł asocjacyjnych	46
7.1	Algorytm zachłanny dla konstruowania częściowych reguł asocjacyjnych	46
7.2	Algorytmy zachłanne z wagami dla konstruowania częściowych reguł asocjacyjnych	50
7.3	Podejście oparte na dynamicznym programowaniu dla optymalizacji reguł asocjacyjnych	53
7.4	Podejście oparte na dynamicznym programowaniu dla konstruowania systemów reguł asocjacyjnych	55
7.5	Porównanie heurystyk dla optymalizacji reguł asocjacyjnych	59
8	Podsumowanie	62
9	Omówienie pozostałych osiągnięć naukowo-badawczych	63

4 Podstawowe osiągnięcie

Jako podstawowe osiągnięcie naukowe przedstawiam cykl dziesięciu publikacji monotematycznych pt.: “**Badanie i optymalizacja reguł decyzyjnych i asocjacyjnych**”, na które składają się: monografia [41] (pozycja 1), której jestem współautorem, cztery wysokopunktowane artykuły [9, 10, 38, 39] z listy JCR (pozycje 2, 5, 6, 7), jeden artykuł z czasopisma Transactions on Rough Sets [70] (pozycja 3) oraz cztery publikacje z materiałów konferencyjnych [65, 67, 4, 5] (pozycje 4, 8, 9, 10). Publikacje wchodzące w skład podstawowego osiągnięcia naukowego zostały podane w porządku tematycznym.

1. Moshkov, M., Zielosko, B.: Combinatorial Machine Learning: A Rough Set Approach. Studies in Computational Intelligence, vol. 360. Springer, Heidelberg (2011) [40%]
2. Amin, T., Chikalov, I., Moshkov, M., Zielosko, B.: Dynamic programming approach to optimization of approximate decision rules. Inf. Sci. 221, 403-418 (2013) [35%]
3. Zielosko, B.: Sequential optimization of γ -decision rules relative to length, coverage and number of misclassifications. Transactions on Rough Sets 19, LNCS, vol. 8988, 65-82, Springer (2015)
4. Zielosko, B.: Optimization of approximate decision rules relative to coverage. In: Proc. 10th International Conference Beyond Databases, Architectures, and Structures (Ustroń, Poland), CCIS, vol. 424, 170-179, Springer (2014) (Best paper award)
5. Amin, T., Chikalov, I., Moshkov, M., Zielosko, B.: Relationships between length and coverage of decision rules. Fundam. Inform. 129, 1-13 (2014) [30%]
6. Moshkov, M., Piliszczuk, M., Zielosko, B.: Greedy algorithm for construction of partial association rules. Fundam. Inform. 92, 259-277 (2009) [55%]
7. Moshkov, M., Piliszczuk, M., Zielosko, B.: Greedy algorithms with weights for construction of partial association rules. Fundam. Inform. 94, 101-120 (2009) [52%]
8. Zielosko, B.: Global optimization of exact association rules relative to coverage. In: Proc. 6th International Conference on Pattern Recognition and Machine Intelligence, Warsaw, Poland (2015), LNCS, vol. 9124, 428-437, Springer (2015)
9. Alsolami, F., Amin, T., Chikalov, I., Moshkov, M., Zielosko, B.: Dynamic programming approach for construction of association rule systems. In: Proc. 24th Concurrency, Specification, and Programming Workshop (Rzeszów, Poland), 12-21, CEUR-WS.org (2015) [25%]
10. Alsolami, F., Amin, T., Moshkov, M., Zielosko, B.: Comparison of heuristics for optimization of association rules. In: Proc. 24th Concurrency, Specification, and Programming Workshop (Rzeszów, Poland), 4-11, CEUR-WS.org (2015) [35%]

Przedstawione wyniki badań, wchodzące w skład podstawowego osiągnięcia, stanowią przyczynek do rozwoju teorii zbiorów przybliżonych oraz jej zastosowań.

4.1 Wprowadzenie

Przedstawiony autoreferat pracy habilitacyjnej dotyczy badania i optymalizacji reguł decyzyjnych i asocjacyjnych, w kontekście teorii zbiorów przybliżonych. Reguły decyzyjne jak i reguły asocjacyjne są jedną z technik odkrywania wiedzy z danych. Są one stosowane do konstrukcji klasyfikatorów oraz w celu reprezentacji i analizy wiedzy zawartej w zbiorach danych [47, 52, 57, 59]. Moje zainteresowania dotyczą m.in. zastosowania reguł jako sposobu reprezentacji wiedzy. Zatem, istotne jest aby konstruować krótkie reguły o stosunkowo dobrej jakości, które pokrywają wiele obiektów. Niestety, zarówno problem minimalizacji długości reguł, jak i problem maksymalizacji pokrycia, są problemami NP-trudnymi [16, 36]. W związku z tym, tematyka prac, których jestem autorem lub współautorem, dotyczy badania algorytmów aproksymacyjnych o stosunkowo niskiej złożoności czasowej dla optymalizacji reguł. Poza efektywnymi, ale przybliżonymi algorytmami, badane są także algorytmy dokładne dla optymalizacji reguł - dla potrzeb naukowych.

Istnieje wiele podejść dla konstruowania reguł decyzyjnych: podejście “brute-force” stosowane dla tablic ze stosunkowo niewielką liczbą atrybutów, podejście Boolean reasoning [42], podejście dziel i zwyciężaj (algorytmy wykorzystujące procedurę sekwencyjnego pokrywania) [15, 22], algorytmy oparte na konstruowaniu drzew decyzyjnych [49], różne rodzaje algorytmów zachłanych [36].

Istnieje także wiele podejść dla konstruowania reguł asocjacyjnych. Jednym z najbardziej popularnych jest algorytm Apriori wykorzystujący tzw. zbiory częste [1]. W ciągu lat zostało opracowanych wiele algorytmów opartych np. na partycjonowaniu danych [50], redukcji zbioru transakcji [2], algorytmy wykorzystujące pionowy format danych [17] i inne [25, 43, 60].

Algorytmy autorskie (narzędzia do prowadzonych badań)

Poniżej zostały krótko przedstawione algorytmy rozumiane jako narzędzia, które powstały w ramach pracy habilitacyjnej i które służą do badania i optymalizacji reguł decyzyjnych i asocjacyjnych.

Pośród algorytmów aproksymacyjnych dla optymalizacji reguł badane były algorytmy zachłanne. W przypadku reguł decyzyjnych, algorytmy zachłanne były badane w mojej pracy doktorskiej (monografia [36]) oraz artykułach [33–35, 37, 46, 63, 73]. W autoreferacie przedstawione zostały wyniki dotyczące algorytmów zachłanych tylko dla reguł asocjacyjnych. Wyniki te dotyczą dokładności algorytmów zachłanych z wagami oraz bez wag dla konstruowania reguł, dolnych granic minimalnej złożoności reguł na podstawie danych uzyskanych podczas pracy algorytmu zachłanego, analizy tzw. 0.5-hipotezy przedstawiającej dobre zachowanie algorytmów zachłanych dla większości tablic decyzyjnych, itp.

Pośród algorytmów dokładnych dla optymalizacji reguł decyzyjnych i asocjacyjnych badane było podejście oparte na dynamicznym programowaniu i utworzone zostały jego rozszerzenia, które pozwalają opisać zbiór optymalnych reguł, wykonać wielostopniową optymalizację względem różnych kryteriów oraz badać

zależność pomiędzy długością a pokryciem reguł, itp. W przypadku reguł asocjacyjnych, zaproponowane algorytmy pozwalają pracować zarówno ze spójnymi jak i niespójnymi danymi. Teoria zbiorów przybliżonych zapewnia narzędzia do pracy z niespójnymi tablicami decyzyjnymi [48].

Zastosowania

Poniżej, zostało krótko opisanych pięć zastosowań dla zaproponowanych algorytmów (narzędzi).

1. Sekwencyjna optymalizacja reguł względem wybranej kombinacji funkcji kosztu: długość, pokrycie i liczba błędnych zaklasyfikowań, pozwala uzyskać określoną liczbę stosownych reguł tylko dla wybranych wierszy (obiektów). W ten sposób, można uniknąć konstruowania ogromnej liczby stosownych reguł, jak z wykorzystaniem algorytmu Apriori [1].
2. Sekwencyjna optymalizacja reguł pozwala konstruować dla danego wiersza reguły całkowicie optymalne względem wybranej kombinacji funkcji kosztu np. optymalne równocześnie względem pokrycia, długości i liczby błędnych zaklasyfikowań, jeśli takie reguły istnieją. Wyniki eksperymentalne pokazują, że dla wielu zbiorów danych, całkowicie optymalne reguły istnieją dla wszystkich lub dla większości obiektów (wierszy) i mają dobre pokrycie, małą liczbę błędnych zaklasyfikowań i są krótkie. Reguły te mogą zostać wykorzystane zarówno jako sposób reprezentacji wiedzy jak i do konstrukcji klasyfikatorów (zobacz kolejne zastosowanie).
3. Porównanie dokładności klasyfikatorów regułowych opartych na jednokryterialnej i dwukryterialnej optymalizacji reguł pokazuje, że klasyfikatory oparte na regułach zoptymalizowanych najpierw względem pokrycia, a następnie względem długości lub najpierw względem długości, a następnie względem pokrycia, dają często lepsze wyniki, niż klasyfikatory oparte na regułach zoptymalizowanych tylko względem długości lub tylko względem pokrycia.
4. Porównanie pięciu heurystyk zachłanych dla konstruowania reguł asocjacyjnych, pod kątem długości i pokrycia konstruowanych reguł. Wygenerowane reguły zostały także porównane z regułami optymalnymi, skonstruowanymi za pomocą algorytmów dynamicznego programowania. Średnia relatywna różnica odnośnie długości skonstruowanych reguł z wykorzystaniem najlepszej heurystyki i minimalnej długości reguł wynosi najwyżej 4%. Taka sama sytuacja występuje w przypadku pokrycia reguł. W związku z tym, jako wynik, zostały wybrane dwie różne heurystyki dla minimalizacji długości reguł oraz dla maksymalizacji pokrycia reguł.
5. Opracowanie algorytmu, który konstruuje, dla danego zbioru danych, system reguł asocjacyjnych o stosunkowo dobrej jakości, które pokrywają większą część obiektów (wierszy). Dla większości analizowanych zbiorów danych, liczność konstruowanego zbioru reguł jest rozsądna. Opracowany algorytm symuluje algorytm zachłanny dla problemu pokrycia zbioru tzn., w każdej iteracji wybiera regułę, która pokrywa maksymalną liczbę niepokrytych wcześniej wierszy (obiektów). Fakt ten pozwala uzyskać nietrywialne dolne grani-

ce minimalnej liczebności systemów reguł, na podstawie danych uzyskanych podczas pracy algorytmu.

Wyniki teoretyczne

Autoreferat rozprawy habilitacyjnej zawiera wiele wyników teoretycznych. Wiele z nich dotyczy złożoności rozważanych problemów oraz poprawności i złożoności zaproponowanych algorytmów. Przedstawiono wyniki opisujące sytuację dla prawie wszystkich tablic decyzyjnych oraz wyniki opisujące tzw. 0.5-hipotezę dla algorytmów zachłanych.

Znaczna część wyników teoretycznych dotyczy porównania drzew decyzyjnych, systemów reguł decyzyjnych oraz testów (super reduktów): badanie powiązań pomiędzy systemami reguł, drzewami i testami; porównanie górnych i dolnych granic złożoności drzew, systemów reguł i testów dla tablic decyzyjnych; badanie złożoności czasowej algorytmów zachłanych i dynamicznego programowania dla konstruowania reguł. Przedstawiono także porównanie złożoności czasowej drzew i systemów reguł dla skończonych i nieskończonych systemów informacyjnych oraz porównanie drzew i systemów reguł jako algorytmów dla problemów optymalizacji kombinatorycznej.

Struktura pracy

Wyniki badań zostały przedstawione w trzech rozdziałach.

W rozdziale 5 zostały opisane wyniki badań zawarte w książce [41]. Dotyczą one, w szczególności, porównania drzew decyzyjnych, reguł i testów (super-reduktów). Był to punkt początkowy dla różnych kierunków badań, m.in. badania podejścia opartego na dynamicznym programowaniu dla reguł oraz badania reguł decyzyjnych i testów dla tablic decyzyjnych z wielowartościową decyzją (wiersze tablicy mają przypisany zbiór wartości atrybutu decyzyjnego).

W rozdziale 6 zostały opisane wyniki badań, na podstawie artykułów [9, 10, 65, 70]. Rozdział ten dotyczy rozszerzeń podejścia opartego na dynamicznym programowaniu dla analizy i optymalizacji reguł decyzyjnych. Wymienione wcześniej trzy pierwsze zastosowania zostały opisane w tym rozdziale.

W rozdziale 7 zostało opisane podejście zachłanne oraz podejście oparte na dynamicznym programowaniu dla optymalizacji reguł asocjacyjnych. Rozdział ten zawiera wyniki badań na podstawie artykułów [4, 5, 38, 39, 67]. Wymienione wcześniej dwa ostatnie zastosowania zostały opisane w tym rozdziale.

5 Porównanie reguł decyzyjnych, drzew i testów

W rozdziale tym zostaną przedstawione wyniki badań opublikowane w książce [41] napisanej przez M. Moshkova i B. Zielosko. Książka ta może być traktowana jak mieszanka monografii naukowej oraz skryptu zawierającego notatki do wykładów. Autorzy próbowali usystematyzować narzędzia do pracy z dokładnymi i przybliżonymi drzewami decyzyjnymi, regułami i testami, zarówno dla tablic

decyzyjnych z jednowartościową decyzją (wiersze tablicy mają przypisaną jedną wartość atrybutu decyzyjnego), jak i dla tablic decyzyjnych z wielowartościową decyzją (wiersze tablicy mają przypisany zbiór wartości atrybutu decyzyjnego). W celu wypełnienia różnych luk, które pojawiły się w trakcie prac dotyczących uszeregowania wyników badań, autorzy dodali wiele nieopublikowanych wcześniej wyników.

Pierwsze wykłady związane z tematyką zawartą w książce były prowadzone w Rosji w 1984 roku przez prof. M. Moshkova. Tematyka wykładu dotyczyła zagadnień związanych z drzewami decyzyjnymi i testami dla tablic decyzyjnych z jednowartościową decyzją. W 2005 roku w Polsce, zagadnienia związane z przybliżonymi drzewami i super-reduktami oraz tablicami decyzyjnymi z wielowartościową decyzją zostały dodane do nowej wersji tego wykładu. Po opublikowaniu wielu artykułów dotyczących częściowych pokryć, reduktów, reguł decyzyjnych i asocjacyjnych [33–35, 37–39, 46, 63, 64, 73, 74] wraz z monografią [36], autorzy zdecydowali się dodać tematykę reguł decyzyjnych do prowadzonego wykładu.

Książka ta stanowi istotne rozszerzenie przedmiotu Combinatorial Machine Learning prowadzonego w King Abdullah University of Science and Technology (KAUST) w Arabii Saudyjskiej, który został wspólnie przygotowany i prowadzony przez dwa lata przez M. Moshkova i B. Zielosko. Książka “Combinatorial Machine Learning. A Rough Set Approach” staje się coraz bardziej popularna w ciągu ostatnich lat, co potwierdzają raporty publikowane przez wydawnictwo Springer. Liczba ściągniętych rozdziałów wynosiła 389 w 2011 roku, 691 w 2012 roku, 1874 w 2013 roku i 2611 w 2014 roku.

Jednym z głównych celów tej książki jest porównanie złożoności czasowej drzew decyzyjnych, super-reduktów oraz systemów reguł decyzyjnych. W związku z tym, w rozdziale zostaną przedstawione nie tylko wyniki badań dotyczące reguł ale także pewne wyniki dotyczące drzew i testów. Zawarte w książce wyniki dotyczące reguł decyzyjnych i systemów reguł decyzyjnych zostały uzyskane przez B. Zielosko.

Struktura tego rozdziału jest bardzo podobna do struktury książki. Pięć pierwszych rozdziałów dotyczy badania powiązań pomiędzy drzewami decyzyjnymi, regułami i reduktami, analizy granic ich złożoności oraz opracowania algorytmów dla konstruowania reguł. Trzy ostatnie rozdziały dotyczą zastosowań stworzonych algorytmów (narzędzi) dla nadzorowanego uczenia maszynowego, porównania złożoności czasowej drzew decyzyjnych i systemów reguł dla skończonych i nieskończonych systemów informacyjnych oraz problemów optymalizacji kombinatorycznej.

5.1 Powiązania pomiędzy regułami decyzyjnymi, drzewami i testami

W rozdziale zostaną przedstawione wyniki badań dla tablic decyzyjnych z jednowartościową decyzją. Dla uproszczenia, rozważane są głównie tablice decyzyjne zawierające tylko binarne atrybuty warunkowe. Zostaną przedstawione definicje pojęć tablicy decyzyjnej, drzewa, reguły, systemu reguł oraz testu (super-reduktu) i reduktu, oraz powiązania pomiędzy drzewami, regułami i testami.

Tablice decyzyjne, drzewa, reguły i testy

Tablica decyzyjna jest tabelą której elementy należą do zbioru $\{0, 1\}$. Kolumny tabeli są oznaczone przez atrybuty f_1, \dots, f_n . Wiersze tabeli są parami różne i każdy wiersz ma przypisaną liczbę naturalną (decyzję) interpretowaną jako wartość atrybutu decyzyjnego. Jest to tabela z jednowartościową decyzją. Formalnie, tablicę decyzyjną można zdefiniować jako $T = (U, A \cup \{d\})$, gdzie $U = \{r_1, \dots, r_m\}$ jest niepustym, skończonym zbiorem obiektów (wierszy), $A = \{f_1, \dots, f_n\}$ jest niepustym, skończonym zbiorem atrybutów, $f : U \rightarrow V_f$ jest funkcją dla dowolnego $f \in A$, V_f jest zbiorem wartości atrybutu f . Elementy zbioru A są nazywane atrybutami warunkowymi, $d \notin A$ jest wyróżnionym atrybutem, nazywanym atrybutem decyzyjnym.

Drzewo decyzyjne nad T to skończone drzewo posiadające korzeń. W drzewie tym każdy węzeł końcowy (liść) jest oznaczony przez decyzję (liczbę naturalną), każdy niekończący węzeł (nazywany *pracującym*) jest oznaczony przez atrybut ze zbioru $\{f_1, \dots, f_n\}$. Z każdego niekończącego węzła wychodzą dwie krawędzie oznaczone odpowiednio przez wartości 0 lub 1.

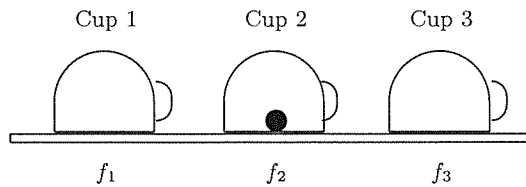
Niech Γ będzie drzewem decyzyjnym nad T . Dla danego wiersza r tabeli T drzewo zaczyna pracę od korzenia. Jeśli rozważany węzeł drzewa jest węzłem końcowym, wówczas wynikiem pracy drzewa Γ jest liczba przypisana do tego węzła. Niech rozważany węzeł będzie niekończący, oznaczony przez atrybut f_i . Jeśli wartość atrybutu f_i w rozważanym wierszu jest 0, wówczas należy przejść wzdłuż krawędzi oznaczonej przez 0. W przeciwnym przypadku, należy przejść wzdłuż krawędzi oznaczonej przez 1, itd.

Powiemy, że Γ jest *drzewem decyzyjnym dla T* jeśli, dla dowolnego wiersza r z T , drzewo Γ kończy pracę w węźle końcowym, który jest oznaczony przez decyzję przypisaną do rozważanego wiersza.

Przez $h(\Gamma)$ jest oznaczana *głębokość Γ* . Jest to maksymalna długość ścieżki od korzenia do węzła końcowego. Przez $h(T)$ jest oznaczana minimalna głębokość drzewa decyzyjnego dla tabeli T . Drzewa decyzyjne są traktowane jako algorytmy pracujące w sposób sekwencyjny.

Reguła decyzyjna nad T to wyrażenie postaci $f_{i_1} = b_1 \wedge \dots \wedge f_{i_m} = b_m \rightarrow t$ gdzie $f_{i_1}, \dots, f_{i_m} \in \{f_1, \dots, f_n\}$, $b_1, \dots, b_m \in \{0, 1\}$ i $t \in \mathbb{N}$. Liczba m oznacza *długość reguły*. Powiemy, że reguła ta jest *zrealizowana* dla wiersza $r = (\delta_1, \dots, \delta_n)$ jeśli $\delta_{i_1} = b_1, \dots, \delta_{i_m} = b_m$. Reguła jest *prawdziwa dla T* jeśli dla dowolnego wiersza r z T , dla którego reguła jest zrealizowana, wiersz r ma przypisaną decyzję t . Rozważana reguła jest *regułą dla T i r* , jeśli jest prawdziwa dla T i zrealizowana dla r . Przez $L(T, r)$ jest oznaczana minimalna długość reguły dla T i r . Reguła decyzyjna dla T i r jest *nieskracalna*, jeśli dowolna reguła uzyskana przez usunięcie pary "atrybut=wartość" z lewej strony rozważanej reguły, nie jest regułą dla T i r .

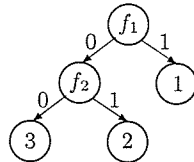
System reguł decyzyjnych S nad T jest niepustym, skończonym zbiorem reguł nad T . System S jest nazywany *kompletnym systemem reguł decyzyjnych dla T* , jeśli każda reguła z S jest prawdziwa dla T i dla każdego wiersza z T istnieje reguła z S , która jest zrealizowana dla tego wiersza. Przez $L(S)$ jest oznaczana maksymalna długość reguły z S , przez $L(T)$ jest oznaczana minimalna wartość



Rysunek 1

f_1	f_2	f_3	
1	0	0	1
0	1	0	2
0	0	1	3

Rysunek 2



Rysunek 3

$\{f_1, f_2, f_3\}$
 $\{f_1, f_2\}$
 $\{f_1, f_3\}$
 $\{f_2, f_3\}$

Rysunek 4

$L(S)$ spośród wszystkich kompletnych systemów reguł S dla T . Rozważanie tej miary złożoności dla systemów reguł decyzyjnych związane jest z założeniem, że systemy reguł decyzyjnych mogą być traktowane jako algorytmy, które pracują w sposób równoległy.

Test (super-redukt) dla T to podzbiór kolumn tabeli T taki, że na przecięciu z tymi kolumnami dowolne dwa wiersze o różnych decyzjach są różne. *Redukt dla T* jest testem dla T takim, że żaden właściwy podzbiór nie jest testem. Oczywiście jest, że każdy test posiada redukt jako podzbiór. Przez $R(T)$ jest oznaczana minimalna liczebność reduktu dla T .

Przykład 1. Przykład związany jest z problemem, który przedstawia Rysunek 1 - trzy odwrócone kubki na stole i mała piłka pod jednym z nich. Należy znaleźć numer kubka pod którym znajduje się piłka. W tym celu zostały użyte atrybuty f_i , $i = 1, 2, 3$. Po podniesieniu i -tego kubka sprawdzamy czy znajduje się tam piłka. Jeśli tak, wartość atrybutu f_i wynosi 1. W przeciwnym przypadku, wartość atrybutu f_i wynosi 0. Tablica decyzyjna T odpowiadająca rozważanemu problemowi została przedstawiona na Rysunku 2.

Drzewo decyzyjne dla T o minimalnej głębokości przedstawia Rysunek 3. Rysunek 4 przedstawia wszystkie testy dla tabeli T . Oczywiście jest, że $h(T) = 2$ i $R(T) = 2$.

Można pokazać, że $\{f_1 = 1 \rightarrow 1, f_2 = 1 \rightarrow 2, f_3 = 1 \rightarrow 3\}$ jest kompletnym systemem reguł decyzyjnych dla T i dla $i = 1, 2, 3$, i -ta reguła jest najkrótszą regułą, która jest prawdziwa dla T i zrealizowana dla i -tego wiersza tabeli T . Zatem, $L(T) = 1$.

Powiązania pomiędzy drzewami decyzyjnymi, regułami i testami

Twierdzenie 1. Niech T będzie tablicą decyzyjną posiadającą n kolumn oznaczonych przez atrybuty f_1, \dots, f_n .

1. Jeśli Γ jest drzewem decyzyjnym dla tabeli T wówczas zbiór atrybutów występujących w niekońcowych węzłach drzewa Γ jest testem dla tabeli T .
2. Niech $F = \{f_{i_1}, \dots, f_{i_m}\}$ będzie testem dla T . Wówczas istnieje drzewo decyzyjne Γ dla T , w którym występują atrybuty tylko z F i dla którego $h(\Gamma) = m$.

Wniosek 1. Niech T będzie tablicą decyzyjną. Wówczas $h(T) \leq R(T)$.

Twierdzenie 2. Niech T będzie tablicą decyzyjną z n kolumnami oznaczonymi przez atrybuty f_1, \dots, f_n .

1. Jeśli S jest kompletnym systemem reguł decyzyjnych dla T , wówczas zbiór atrybutów występujących w regułach z S jest testem dla T .
2. Jeśli $F = \{f_{i_1}, \dots, f_{i_m}\}$ jest testem dla T , wówczas istnieje kompletny system S reguł decyzyjnych dla T w którym występują atrybuty tylko z F i dla których $L(S) = m$.

Wniosek 2. $L(T) \leq R(T)$.

Niech Γ będzie drzewem decyzyjnym dla T i τ będzie ścieżką w Γ prowadzącą od korzenia do węzła końcowego, w której niekońcowe węzły są oznaczone przez atrybuty f_{i_1}, \dots, f_{i_m} , krawędzie są oznaczone przez liczby b_1, \dots, b_m i węzeł końcowy τ ma przypisaną decyzję t . Ścieżce τ odpowiada reguła decyzyjna $\text{rule}(\tau): f_{i_1} = b_1 \wedge \dots \wedge f_{i_m} = b_m \rightarrow t$.

Twierdzenie 3. Niech Γ będzie drzewem decyzyjnym dla T i S będzie zbiorem reguł decyzyjnych odpowiadających ścieżkom w Γ od korzenia do węzłów końcowych. Wówczas S jest kompletnym systemem reguł decyzyjnych dla T i $L(S) = h(\Gamma)$.

Wniosek 3. $L(T) \leq h(T)$.

Wszystkie wyniki opisane w tym rozdziale mogą zostać rozszerzone dla k -wartościowych tablic decyzyjnych wypełnionych przez liczby ze zbioru $\{0, 1, \dots, k-1\}$.

5.2 Granice złożoności reguł decyzyjnych, drzew i testów

Rozdział ten jest kontynuacją badań dla tablic decyzyjnych z jednowartościową decyzją. Poniżej zostaną przedstawione granice złożoności nie tylko dla reguł ale także dla drzew i testów, w celu pokazania ich relacji w stosunku do granic złożoności dla reguł decyzyjnych.

Z Wniosków 1 i 3 wynika, że $L(T) \leq h(T) \leq R(T)$. Zatem każda dolna granica $L(T)$ jest także dolną granicą $h(T)$ i $R(T)$ i każda dolna granica $h(T)$ jest także dolną granicą $R(T)$.

Poniższe twierdzenia dotyczą dolnej granicy $h(T)$ i konsekwentnie, dolnej granicy $R(T)$. Przez $D(T)$ jest oznaczona liczba różnych decyzji w tablicy decyzyjnej T .

Twierdzenie 4. Niech T będzie niepustą tablicą decyzyjną. Wówczas $h(T) \geq \log_2 D(T)$.

Twierdzenie 5. Niech T będzie tablicą decyzyjną. Wówczas $h(T) \geq \log_2(R(T)+1)$.

Niech T będzie tablicą decyzyjną posiadającą n kolumn oznaczonych przez atrybuty f_1, \dots, f_n . Podtabela tablicy T to tabela uzyskana z T przez usunięcie pewnych wierszy. Niech $\{f_{i_1}, \dots, f_{i_m}\} \subseteq \{f_1, \dots, f_n\}$ i $\delta_1, \dots, \delta_m \in \{0, 1\}$. Przez $T(f_{i_1}, \delta_1) \dots (f_{i_m}, \delta_m)$ jest oznaczana podtabela tabeli T , która posiada wiersze, które na przecięciu z kolumnami f_{i_1}, \dots, f_{i_m} posiadają liczby $\delta_1, \dots, \delta_m$. Tabela T jest nazywana zdegenerowaną tabelą, jeśli T nie posiada wierszy lub wszystkie wiersze w T mają przypisaną tą samą decyzję.

Poniżej zostanie zdefiniowany parametr $M(T)$ tabeli T . Jeśli T jest zdegenerowana wówczas $M(T) = 0$. Niech T będzie niezdegenerowana. Niech $\bar{\delta} = (\delta_1, \dots, \delta_n) \in \{0, 1\}^n$. Przez $M(T, \bar{\delta})$ jest oznaczana minimalna liczba naturalna m taka, że istnieją atrybuty $f_{i_1}, \dots, f_{i_m} \in \{f_1, \dots, f_n\}$ dla których $T(f_{i_1}, \delta_{i_1}) \dots (f_{i_m}, \delta_{i_m})$ jest zdegenerowaną tabelą. Wówczas

$$M(T) = \max\{M(T, \bar{\delta}) : \bar{\delta} \in \{0, 1\}^n\}.$$

Twierdzenie 6. Niech T będzie tablicą decyzyjną. Wówczas $h(T) \geq M(T)$.

Można znaleźć dokładną formułę dla wartości $L(T)$ korzystając z parametru $M(T, \bar{\delta})$.

Twierdzenie 7. Niech T będzie tablicą decyzyjną i $\text{Row}(T)$ będzie zbiorem wierszy z T . Wówczas, $L(T, \bar{\delta}) = M(T, \bar{\delta})$ dla dowolnego wiersza $\bar{\delta} \in \text{Row}(T)$ i $L(T) = \max\{M(T, \bar{\delta}) : \bar{\delta} \in \text{Row}(T)\}$.

Wniosek 4. $L(T) \leq M(T) \leq h(T) \leq R(T)$.

Poniżej zostanie przedstawiona górna granica wartości $R(T)$. Jest to także górna granica wartości $L(T)$ i $h(T)$. Przez $N(T)$ jest oznaczana liczba wierszy w tabeli T .

Twierdzenie 8. Niech T będzie tablicą decyzyjną. Wówczas $R(T) \leq N(T) - 1$.

Poniżej zostanie przedstawiona górna granica wartości $h(T)$. Jest to także górna granica wartości $L(T)$.

Twierdzenie 9. Niech T będzie tablicą decyzyjną. Wówczas $h(T) \leq M(T) \log_2 N(T)$.

Można analizować nie tylko binarne tablice decyzyjne wypełnione przez liczby ze zbioru $\{0, 1\}$ ale także k -wartościowe tablice wypełnione przez liczby ze zbioru $\{0, 1, \dots, k-1\}$, gdzie $k > 2$. Dla takich tablic, wszystkie wyniki przedstawione powyżej są prawdziwe, z wyjątkiem Twierdzenia 4 i Twierdzenia 5.

Dla k -wartościowych tablic, zamiast granic $h(T) \geq \log_2 D(T)$ i $h(T) \geq \log_2(R(T)+1)$, istnieją granice $h(T) \geq \log_k D(T)$ i $h(T) \geq \log_k((k-1)R(T)+1)$.

Prosta analiza górnych granic $R(T)$ i $h(T)$ (Twierdzenia 8 i 9) pokazuje, że w ogólnym przypadku, nie są one wystarczająco dobre jako górne granice $L(T)$. Dolne granice $h(T)$ nie są, w ogólnym przypadku, dolnymi granicami $L(T)$.

5.3 Algorytmy dla konstruowania reguł decyzyjnych

W rozdziale tym zostaną przedstawione dwa problemy optymalizacji związane z tablicami decyzyjnymi. *Problem minimalizacji długości reguły decyzyjnej*: dla danej tablicy decyzyjnej T i wiersza r z T należy skonstruować regułę decyzyjną nad T , która jest prawdziwa dla T , zrealizowana dla r i posiada minimalną długość. *Problem optymalizacji systemu reguł decyzyjnych*: dla danej tablicy decyzyjnej T należy skonstruować kompletny system reguł decyzyjnych S dla T o minimalnej wartości parametru $L(S)$.

Poniżej zostaną przedstawione wyniki badań pokazujące, że te problemy są NP -trudne oraz wyniki badań dotyczące dokładności aproksymacyjnych algorytmów wielomianowych dla rozwiązania tych problemów. Zostaną także przedstawione algorytmy zachłanne (aproksymacyjne) i algorytmy dynamicznego programowania (dokładne) dla rozwiązania tych problemów.

Algorytmy aproksymacyjne dla optymalizacji reguł decyzyjnych

Niech T będzie niezdegenerowaną tablicą decyzyjną zawierającą n kolumn oznaczonych przez atrybuty f_1, \dots, f_n , i $r = (b_1, \dots, b_n)$ będzie wierszem tabeli T z przypisaną decyzją t . Przez $A(T, r)$ jest oznaczany zbiór wszystkich wierszy z tabeli T , które mają przypisaną decyzję inną niż t . Atrybut f_i *separuje* od wiersza r wiersz r' z $A(T, r)$, jeśli r różni się od r' na przecięciu z kolumną f_i .

Algorytm 1 przedstawia *algorytm zachłanny dla konstruowania reguły decyzyjnej* dla T i r . Wynikiem działania algorytmu jest reguła decyzyjna dla T

Algorytm 1: Algorytm zachłanny dla konstruowania reguły decyzyjnej dla T i r .

Wejście: Tablica decyzyjna T z atrybutami f_1, \dots, f_n , wiersz $r = (b_1, \dots, b_n)$ w T z przypisaną decyzją t .
Wyjście: Reguła decyzyjna dla T i r .
 $Q \leftarrow \emptyset$;
while atrybuty z Q separują od r mniej niż $|A(T, r)|$ wierszy z $A(T, r)$ **do**
 | wybierz $f_i \in \{f_1, \dots, f_n\}$ o minimalnym indeksie i taki, że f_i separuje od r
 | maksymalną liczbę wierszy z $A(T, r)$ nieodseparowanych przez atrybuty z Q ;
 | $Q \leftarrow Q \cup \{f_i\}$;
end
return $\bigwedge_{f_i \in Q} (f_i = b_i) \rightarrow t$;

i r : $f_{i_1} = b_{i_1} \wedge \dots \wedge f_{i_m} = b_{i_m} \rightarrow t$. Przez $L_{\text{greedy}}(T, r)$ jest oznaczana długość konstruowanej reguły, przez $P(T, r)$ jest oznaczana liczba $|A(T, r)|$.

Twierdzenie 10. Niech T będzie niezdegenerowaną tablicą decyzyjną i r będzie wierszem tabeli T . Wówczas $L_{\text{greedy}}(T, r) \leq L(T, r) \ln P(T, r) + 1$.

Przedstawiony algorytm może zostać wykorzystany dla skonstruowania kompletnego systemu reguł decyzyjnych dla T . W tym celu, algorytm ten jest stosowany sekwencyjnie do tabeli T i każdego wiersza r z T . Wynikiem jest system reguł S , w którym każda reguła jest prawdziwa dla T i dla każdego wiersza z T istnieje reguła w S , która jest zrealizowana dla tego wiersza.

$L_{\text{greedy}}(T) = L(S)$ i $K(T) = \max\{P(T, r) : r \in \text{Row}(T)\}$, gdzie $\text{Row}(T)$ jest zbiorem wierszy z T . Oczywiście jest, że $L(T) = \max\{L(T, r) : r \in \text{Row}(T)\}$. Korzystając z Twierdzenia 10 można uzyskać

Twierdzenie 11. *Niech T będzie niezdegenerowaną tablicą decyzyjną. Wówczas*

$$L_{\text{greedy}}(T) \leq L(T) \ln K(T) + 1.$$

Poniższe twierdzenia także zostały udowodnione.

Twierdzenie 12. *Problem minimalizacji długości reguły decyzyjnej jest NP-trudny.*

Twierdzenie 13. *Jeśli $NP \not\subseteq DTIME(n^{O(\log \log n)})$ wówczas, dla dowolnego ε , $0 < \varepsilon < 1$, nie istnieje algorytm o złożoności wielomianowej taki, że dla danej niezdegenerowanej tablicy decyzyjnej T i wiersza r z T , konstruuje regułę decyzyjną, która jest prawdziwa dla T , zrealizowana dla r i której długość jest najwyżej $(1 - \varepsilon)L(T, r) \ln P(T, r)$.*

Twierdzenie 14. *Problem optymalizacji systemu reguł decyzyjnych jest NP-trudny.*

Twierdzenie 15. *Jeśli $NP \not\subseteq DTIME(n^{O(\log \log n)})$ wówczas, dla dowolnego ε , $0 < \varepsilon < 1$, nie istnieje algorytm o złożoności wielomianowej taki, że dla danej niezdegenerowanej tablicy decyzyjnej T , konstruuje kompletny system reguł S dla T taki, że $L(S) \leq (1 - \varepsilon)L(T) \ln K(T)$.*

Porównanie Twierdzeń 10 i 13 oraz 11 i 15 pokazuje, że przyjmując założenie dotyczące klasy NP , $NP \not\subseteq DTIME(n^{O(\log \log n)})$, rozważane algorytmy są bliskie (z punktu widzenia dokładności) najlepszym wielomianowym algorytmom aproksymacyjnym dla minimalizacji długości reguły oraz dla optymalizacji systemu reguł decyzyjnych.

Podobne wyniki można uzyskać dla drzew decyzyjnych i testów. Jednakże, algorytm zachłanny dla konstruowania drzewa decyzyjnego jest bardziej skomplikowany.

Dokładny algorytm dla optymalizacji reguł decyzyjnych

W tej części rozdziału zostanie przedstawiony dokładny algorytm dla optymalizacji reguł decyzyjnych, oparty na podejściu dynamicznego programowania. Algorytm ten, w najgorszym przypadku, posiada wykładniczą złożoność czasową. Jednak, w dalszej części autoreferatu (kiedy zostanie przedstawione lokalne

podejście do badania drzew decyzyjnych i reguł dla nieskończonych systemów informacyjnych), zostaną opisane nieskończone systemy informacyjne dla których algorytm ten posiada (dla tablic decyzyjnych uzyskanych z rozważanych systemów) wielomianową złożoność czasową zależną od liczby kolumn (atrybutów) w tabeli.

Niepusta podtabela T' tabeli T jest nazywana *separowalną* podtabelą tabeli T , jeśli istnieją atrybuty f_{i_1}, \dots, f_{i_t} z $\{f_1, \dots, f_n\}$ i liczby $\delta_1, \dots, \delta_t$ z $\{0, 1\}$ takie, że $T' = T(f_{i_1}, \delta_1) \dots (f_{i_t}, \delta_t)$. Przez $SEP(T)$ jest oznaczany zbiór wszystkich separowalnych podtabel tabeli T , wraz z tabelą T . Dla dowolnej podtabeli $D \in SEP(T)$, przez $E(D)$ jest oznaczany zbiór atrybutów z $\{f_1, \dots, f_n\}$, które nie są stałe w D (posiadają przynajmniej dwie różne wartości).

Poniżej zostanie opisany algorytm V dla minimalizacji długości reguł decyzyjnych. Praca algorytmu V składa się z dwóch części.

Część pierwsza pracy algorytmu V dotyczy konstruowania zbioru $SEP(T)$.

Krok 1: Ustaw $SEP = \{T\}$ i przejdź do kroku drugiego. Po pierwszym kroku tabela T nie jest oznaczona jako tabela "przetworzona".

Przypuśćmy, że $t \geq 1$ kroków zostało wykonanych.

Krok ($t + 1$): Niech wszystkie tabele w zbiorze SEP będą oznaczone jako tabele "przetworzone". W tej sytuacji, pierwsza część pracy algorytmu V zostaje zakończona i $SEP(T) = SEP$. Niech istnieje tabela $D \in SEP$, która nie została "przetworzona". Do zbioru SEP dodaj wszystkie podtabele typu $D(f_i, \delta)$, gdzie $f_i \in E(D)$ i $\delta \in \{0, 1\}$, które nie były w zbiorze SEP , oznacz tabelę D jako "przetworzoną" i przejdź do kroku ($t + 2$).

Podczas pierwszej części algorytm V wykonuje $|SEP(T)| + 2$ kroków.

Druga część pracy algorytmu V dotyczy konstruowania optymalnych reguł decyzyjnych $V(T, r)$ dla każdego wiersza r z T . Zaczynając od najmniejszych podtabel ze zbioru $SEP(T)$, algorytm V w każdym kroku przypisuje do każdego wiersza r podtabeli D optymalną regułę decyzyjną dla D i r (*optymalna* reguła dla D i r oznacza regułę decyzyjną o minimalnej długości, która jest prawdziwa dla D i zrealizowana dla r).

Przypuśćmy, że $p \geq 0$ kroków drugiej części algorytmu V zostało właśnie wykonanych.

Krok ($p + 1$): Jeśli każdy wiersz r tabeli T ma przypisaną regułę decyzyjną, wówczas reguła przypisana do r jest wynikiem pracy algorytmu V dla T i r (reguła ta jest oznaczana przez $V(T, r)$). W przeciwnym przypadku, wybierz w zbiorze $SEP(T)$ tabelę D spełniającą następujące warunki:

- a) wiersze w D nie mają przypisanych reguł decyzyjnych;
- b) D jest tabelą zdegenerowaną lub tabelą niezdegenerowaną taką, że dla wszystkich separowalnych podtabel tabeli D , postaci $D(f_i, \delta)$, $f_i \in E(D)$, $\delta \in \{0, 1\}$, każdy wiersz ma przypisaną regułę decyzyjną.

Niech D będzie zdegenerowaną tabelą w której wszystkie wiersze mają przypisaną taką samą decyzję d . Wówczas, przypisz do każdego wiersza tabeli D regułę decyzyjną postaci $\rightarrow d$.

Niech D będzie niezdegenerowaną tablicą decyzyjną i $r = (\delta_1, \dots, \delta_n)$ będzie wierszem w D , który ma przypisaną decyzję d . Dla dowolnego $f_i \in E(D)$,

konstruowana jest reguła $rule(r, f_i)$. Niech wiersz r w tabeli $D(f_i, \delta_i)$ ma przypisaną regułę $\alpha_i \rightarrow d$. Wówczas reguła $rule(r, f_i)$ jest równa $f_i = \delta_i \wedge \alpha_i \rightarrow d$. Przypisz do wiersza r tabeli D jedną z reguł $rule(r, f_i)$, $f_i \in E(D)$, która posiada minimalną długość. Reguła ta jest oznaczana przez $V(T, r)$. Reguła jest przypisywana do każdego wiersza tabeli D w ten sam sposób, następnie przejdź do kroku $(p + 2)$. Podczas drugiej części algorytm V wykonuje $|SEP(T)| + 1$ kroków.

Twierdzenie 16. *Dla dowolnej niezdegenerowanej tablicy decyzyjnej T i dla dowolnego wiersza r w T , algorytm V konstruuje regułę decyzyjną $V(T, r)$, która jest prawdziwa dla T , zrealizowana dla r , i posiada minimalną długość $L(T, r)$. Podczas konstruowania optymalnych reguł decyzyjnych dla wierszy tabeli T , algorytm V wykonuje $2|SEP(T)| + 3$ kroków. Czas pracy algorytmu V jest ograniczony z dołu przez $|SEP(T)|$ i ograniczony z góry przez wielomian od $|SEP(T)|$ i od liczby kolumn w tabeli T .*

Wszystkie wyniki przedstawione w tym rozdziale, po kilku naturalnych zmianach są prawdziwe także dla k -wartościowych tablic decyzyjnych, wypełnionych przez liczby ze zbioru $\{0, 1, \dots, k - 1\}$, $k \geq 3$.

5.4 Tablice decyzyjne z wielowartościową decyzją

Tablice decyzyjne z wielowartościową decyzją mają wiele zastosowań [19, 62]. W przeciwieństwie do tablic decyzyjnych z jednowartościową decyzją, każdy wiersz ma przypisany niepusty, skończony zbiór liczb naturalnych (decyzji). Jeśli celem jest znalezienie wszystkich decyzji przypisanych do danego wiersza należy pracować tak, jak z obiektem matematycznym jakim jest tablica decyzyjna z jednowartościową decyzją: wystarczy zakodować różne zbiory decyzji jako różne liczby. Natomiast, kiedy celem jest znalezienie jednej (dowolnej) decyzji ze zbioru decyzji przypisanych do wiersza, sytuacja jest inna.

W teorii zbiorów przybliżonych [47, 51] rozważane są tablice decyzyjne, które posiadają równe wiersze oznaczone przez różne decyzje. Zbiór decyzji przypisanych do równych wierszy nazywany jest *uogólnioną decyzją* dla każdego z tych równych wierszy. Zazwyczaj, dla danego wiersza wyszukiwana jest uogólniona decyzja. Jednak problem znalezienia dowolnej decyzji lub najczęstszej decyzji z decyzji uogólnionej także może być badany.

Poniżej zostaną przedstawione wyniki badań dotyczące reguł decyzyjnych, systemów reguł decyzyjnych oraz pewne komentarze odnośnie drzew decyzyjnych i testów.

Podstawowe pojęcia

Poniżej zostaną przedstawione definicje związane z tablicami decyzyjnymi z wielowartościową decyzją.

Tablica decyzyjna z wielowartościową decyzją to tabela T wypełniona przez liczby ze zbioru $\{0, 1\}$. Kolumny tabeli są oznaczone przez atrybuty f_1, \dots, f_n .

Wiersze tabeli są parami różne i każdy wiersz ma przypisany niepusty, skończony zbiór liczb naturalnych (zbiór decyzji). Powiemy, że T jest *zdegenerowaną* tabelą, jeśli T nie posiada wierszy lub przecięcie zbioru decyzji przypisanych do wierszy z T jest niepuste (w tym przypadku powiemy, że wiersze tabeli T posiadają *wspólną* decyzję). Należy zauważyć, że każda tablica decyzyjna z jednowartościową decyzją może być interpretowana jako tablica decyzyjna z wielowartościową decyzją. Wówczas, każdy wiersz ma przypisany zbiór decyzji, który zawiera tylko jeden element.

Pojęcie *drzewa decyzyjnego nad T* pokrywa się z pojęciem drzewa decyzyjnego nad tablicą decyzyjną z jednowartościową decyzją. Drzewo decyzyjne Γ nad tablicą decyzyjną T jest *drzewem decyzyjnym dla T* , jeśli dla dowolnego wiersza z T drzewo Γ kończy pracę w węźle końcowym, który ma przypisaną liczbę ze zbioru decyzji, przypisanego do rozważanego wiersza. Przez $h(T)$ jest oznaczana minimalna głębokość drzewa decyzyjnego dla tabeli T .

Pojęcia *reguły decyzyjnej nad T* i *reguły decyzyjnej zrealizowanej dla wiersza r w T* pokrywają się z odpowiednimi pojęciami dla tablic decyzyjnych z jednowartościową decyzją. Reguła nad T , która po prawej stronie posiada decyzję t jest *prawdziwa dla T* , jeśli dla dowolnego wiersza r z T takiego, że reguła ta jest zrealizowana dla wiersza r , liczba t należy do zbioru decyzji przypisanych do wiersza r . Przez $L(T, r)$ jest oznaczana minimalna długość reguły nad T , która jest prawdziwa dla T i zrealizowana dla r . Niepusty zbiór S reguł decyzyjnych nad T jest nazywany *kompletnym systemem reguł decyzyjnych dla T* , jeśli każda reguła z S jest prawdziwa dla T i dla każdego wiersza w T istnieje reguła z S , która jest zrealizowana dla tego wiersza. Przez $L(S)$ jest oznaczana maksymalna długość reguły z S , przez $L(T)$ jest oznaczana minimalna wartość $L(S)$ spośród wszystkich kompletnych systemów reguł S dla T .

Testem dla tabeli T jest podzbiór kolumn $\{f_{i_1}, \dots, f_{i_m}\}$ taki, że dla dowolnych liczb $\delta_1, \dots, \delta_m \in \{0, 1\}$, podtabela $T(f_{i_1}, \delta_1) \dots T(f_{i_m}, \delta_m)$ jest zdegenerowana. Pusty zbiór jest testem dla T wtedy i tylko wtedy, jeśli T jest tabelą zdegenerowaną. Należy zauważyć, że pojęcie testu odpowiada pojęciu testu rozważanemu wcześniej dla tablic decyzyjnych z jednowartościową decyzją. *Redukt* dla tabeli T jest testem dla T takim, dla którego każdy właściwy podzbiór nie jest testem. Oczywiście jest, że każdy test posiada redukt jako podzbiór. Przez $R(T)$ jest oznaczana minimalna liczebność reduktu dla T .

Powiązania pomiędzy drzewami, regułami i testami oraz granice ich złożoności

Twierdzenia 1, 2, 3 i Wnioski 1, 2, 3 opisują związki pomiędzy drzewami decyzyjnymi, regułami i testami, które mogą zostać rozszerzone dla przypadku tablic decyzyjnych z wielowartościową decyzją.

Zamiast Twierdzenia 4 istnieje następujące twierdzenie. Niech T będzie niepustą tablicą decyzyjną z wielowartościową decyzją. Niepusty, skończony zbiór B liczb naturalnych jest nazywany *systemem przedstawicieli dla tabeli T* , jeśli dla każdego wiersza w T , zbiór decyzji przypisanych do tego wiersza posiada nie-

puste przecięcie z B . Przez $S(T)$ jest oznaczana minimalna liczebność systemu przedstawicieli dla tabeli T .

Twierdzenie 17. *Niech T będzie niepustą tablicą decyzyjną z wielowartościową decyzyją. Wówczas $h(T) \geq \log_2 S(T)$.*

Definicje parametrów $M(T)$ i $M(T, \bar{\delta})$ pokrywają się z odpowiednimi definicjami dla tablic decyzyjnych z jednowartościową decyzyją. Twierdzenia 5, 6, 7, 8, 9 i Wniosek 4 opisujące dolne i górne granice $L(T)$, $L(T, \bar{\delta})$, $h(T)$ i $R(T)$ mogą zostać rozszerzone dla przypadku tablic decyzyjnych z wielowartościową decyzyją.

Podobnie jak dla tablic decyzyjnych z jednowartościową decyzyją, dolne i górne granice $h(T)$ i $R(T)$ nie są zbyt użyteczne, w ogólnym przypadku, dla oszacowania $L(T)$.

Aproksymacyjne algorytmy dla optymalizacji reguł decyzyjnych

W tej części rozdziału będą rozważane wielomianowe algorytmy aproksymacyjne dla problemu minimalizacji długości reguły decyzyjnej oraz dla problemu optymalizacji systemu reguł decyzyjnych.

Niech T będzie niezdegenerowaną tabelą z wielowartościową decyzyją zawierającą n kolumn oznaczonych przez atrybuty f_1, \dots, f_n . Niech $r = (b_1, \dots, b_n)$ będzie wierszem tabeli T , $D(r)$ będzie zbiorem decyzyji przypisanych do r i $d \in D(r)$. Przez $A(T, r, d)$ jest oznaczany zbiór wszystkich wierszy r' z T takich, że $d \notin D(r')$.

Algorytm 2 przedstawia *algorytm zachłanny dla konstruowania reguły decyzyjnej dla T , r i d* . Wynikiem działania jest reguła $rule(T, r, d)$ dla T i r : $f_{i_1} = b_{i_1} \wedge \dots \wedge f_{i_m} = b_{i_m} \rightarrow d$.

Algorytm 2: Algorytm zachłanny dla konstruowania reguły decyzyjnej dla T , r i d .

Wejście: Tablica decyzyjna T z wielowartościową decyzyją i atrybutami f_1, \dots, f_n , wiersz $r = (b_1, \dots, b_n)$ w T , $D(r)$ zbiór decyzyji przypisanych do r , $d \in D(r)$.

Wyjście: Reguła decyzyjna dla T , r i d .

$Q \leftarrow \emptyset$;

while atrybuty z Q separują od r mniej niż $|A(T, r, d)|$ wierszy z $A(T, r, d)$ **do**
 wybierz $f_i \in \{f_1, \dots, f_n\}$ o minimalnym indeksie i taki, że f_i separuje od r
 maksymalną liczbę wierszy z $A(T, r, d)$ nieodseparowanych przez atrybuty z Q ;
 $Q \leftarrow Q \cup \{f_i\}$;

end

return $\bigwedge_{f_i \in Q} (f_i = b_i) \rightarrow d$;

Przez $L_{\text{greedy}}(T, r)$ jest oznaczana długość reguły konstruowanej przez następujący algorytm wielomianowy (algorytm ten można nazwać jako *zmodyfikowany algorytm zachłanny*). Dla danej tablicy decyzyjnej T z wielowartościową decyzyją i wiersza r z T , dla każdego $d \in D(r)$, konstruowana jest reguła $rule(T, r, d)$ przez algorytm zachłanny dla konstruowania reguły decyzyjnej dla T , r i d . Spośród

regułę $rule(T, r, d)$, $d \in D(r)$, wybierana jest reguła o minimalnej długości. Reguła ta jest wynikiem działania rozważanego algorytmu. Przez $L_{greedy}(T, r)$ jest oznaczana długość uzyskanej reguły oraz $P(T, r) = \max\{|A(T, r, d)| : d \in D(r)\}$.

Zinodetyfikowany algorytm zachłanny może zostać użyty do skonstruowania kompletnego systemu reguł decyzyjnych dla tablicy decyzyjnej T z wielowartościową decyzją. W tym celu, algorytm ten jest stosowany sekwencyjnie do tabeli T i każdego wiersza r w T . Wynikiem jest system reguł S , w którym każda reguła jest prawdziwa dla T i dla każdego wiersza z T , istnieje reguła z S , która jest zrealizowana dla tego wiersza. $L_{greedy}(T) = L(S)$ i $K(T) = \max\{P(T, r) : r \in Row(T)\}$.

Twierdzenia 10, 11, 13, 15 oraz Twierdzenia 12, 14, opisujące dokładność algorytmów zachłannych i złożoność problemów optymalizacji dla reguł oraz systemów reguł, mogą zostać rozszerzone dla przypadku tablic decyzyjnych z wielowartościową decyzją.

Podobne wyniki można uzyskać dla drzew decyzyjnych i testów. Opierając się one badaniu tzw. granicznych podtabel tablic decyzyjnych. Wówczas można zagwarantować wielomianową złożoność czasową tych algorytmów tylko dla pewnych podzbiorów zbioru tablic decyzyjnych z wielowartościową decyzją.

Dokładne algorytmy dla optymalizacji reguł decyzyjnych

Algorytm V może zostać zastosowany dla dowolnej tablicy decyzyjnej T z wielowartościową decyzją i wiersza r tej tabeli, praktycznie bez żadnych modyfikacji. Należy zauważyć, że w przypadku, kiedy podtabela D jest zdegenerowana, jako d należy wybrać wspólną decyzję dla wszystkich wierszy z D .

Twierdzenie 16 może zostać rozszerzone dla przypadku tablic decyzyjnych z wielowartościową decyzją.

Dla k -wartościowych tablic decyzyjnych wypełnionych przez liczby ze zbioru $\{0, 1, \dots, k-1\}$, $k \geq 3$, wszystkie wyniki przedstawione w tym rozdziale są prawdziwe z wyjątkiem Twierdzenia 17 i rozszerzenia Twierdzenia 5 dla przypadku tablic decyzyjnych z wielowartościową decyzją.

Zamiast granic $h(T) \geq \log_2 S(T)$ i $h(T) \geq \log_2 (R(T)+1)$, dla k -wartościowych tablic istnieją granice $h(T) \geq \log_k S(T)$ i $h(T) \geq \log_k ((k-1)R(T) + 1)$.

5.5 Przybliżone reguły decyzyjne, drzewa i testy

W przypadku zastosowania drzew decyzyjnych, reguł i testów jako sposobu reprezentacji wiedzy, istotne jest aby pracować ze stosunkowo prostymi drzewami, regułami i testami. Jeśli dokładne reguły decyzyjne, drzewa lub testy posiadają dużą złożoność, można rozważać przybliżone drzewa, reguły i testy. W sytuacji kiedy testy, reguły decyzyjne lub drzewa stosowane są jako klasyfikatory, wówczas dokładne testy, reguły decyzyjne i drzewa mogą być przeuczone tzn. zbyt mocno dopasowane do istniejących przykładów lub zbyt mocno zależne od szumu. Wówczas lepiej pracować z przybliżonymi testami, regułami i drzewami. Przybliżone redukty [56, 61], przybliżone reguły decyzyjne [36, 44] i przybliżone drzewa decyzyjne [20, 30, 49] są tematem badań przez wiele ostatnich lat.

Rozdział ten poświęcony jest β -testom, β -drzewom decyzyjnym i β -regułom decyzyjnym, które można traktować jako specjalny rodzaj przybliżonych testów, drzew i reguł.

Podstawowe pojęcia

W rozdziale tym rozważane są tylko tablice decyzyjne z jednowartościową decyzją. Tablice te wypełnione są przez liczby ze zbioru $\{0, 1\}$.

Niech T będzie tablicą decyzyjną z jednowartościową decyzją i T posiada n kolumn oznaczonych przez atrybuty f_1, \dots, f_n . Decyzja, która jest przypisana do maksymalnej liczby wierszy w T nazywana jest *najbardziej wspólną decyzją dla T* . Jeśli istnieje więcej niż jedna taka decyzja, wówczas wybierana jest decyzja o minimalnej wartości. Jeśli T jest pustą tabelą, wówczas 1 jest najbardziej wspólną decyzją dla T . Niech β będzie liczbą rzeczywistą taką, że $0 \leq \beta < 1$. Przez $P(T)$ jest oznaczana liczba nieuporządkowanych par wierszy z T o różnych decyzjach. Jeśli $\beta = 0$ wówczas badane są dokładne drzewa decyzyjne, dokładne testy, dokładne reguły.

Poniżej zostanie zdefiniowane pojęcie β -drzewa decyzyjnego dla T . Niech Γ będzie drzewem decyzyjnym nad T i v będzie węzłem końcowym w Γ . Niech węzły i krawędzie występujące w ścieżce od korzenia do v będą oznaczone odpowiednio przez atrybuty f_{i_1}, \dots, f_{i_m} i liczby $\delta_1, \dots, \delta_m$. Przez $T(v)$ jest oznaczana podtabela $T(f_{i_1}, \delta_1) \dots (f_{i_m}, \delta_m)$ tabeli T . Powiemy, że Γ jest *β -drzewem decyzyjnym dla T* jeśli, dla dowolnego węzła końcowego v w Γ , zachodzi nierówność $P(T(v)) \leq \beta P(T)$, v ma przypisaną najbardziej wspólną decyzję dla $T(v)$ i dla dowolnego wiersza r w T , istnieje węzeł końcowy v w Γ taki, że r należy do tabeli $T(v)$. Przez $h_\beta(T)$ jest oznaczana minimalna głębokość β -drzewa decyzyjnego dla T .

Poniżej zostanie zdefiniowane pojęcie β -testu dla tabeli T . *β -test dla tabeli T* jest podzbiorem kolumn $\{f_{i_1}, \dots, f_{i_m}\}$ takim, że $P(T(f_{i_1}, \delta_1) \dots (f_{i_m}, \delta_m)) \leq \beta P(T)$ dla dowolnych liczb $\delta_1, \dots, \delta_m \in \{0, 1\}$. Zbiór pusty jest β -testem dla T wtedy i tylko wtedy, jeśli T jest tabelą zdegenerowaną. *β -redukt dla tabeli T* jest β -testem dla T takim, że każdy właściwy podzbiór nie jest β -testem. Przez $R_\beta(T)$ jest oznaczana minimalna liczebność β -testu dla tabeli T .

Niech $r = (\delta_1, \dots, \delta_n)$ będzie wierszem z T . Reguła decyzyjna nad T , $f_{i_1} = b_1 \wedge \dots \wedge f_{i_m} = b_m \rightarrow d$ jest nazywana *β -regulą decyzyjną dla T i r* , jeśli $b_1 = \delta_{i_1}, \dots, b_m = \delta_{i_m}$, d jest najbardziej wspólną decyzją dla tabeli $T' = T(f_{i_1}, b_1) \dots (f_{i_m}, b_m)$ i $P(T') \leq \beta P(T)$. Przez $L_\beta(T, r)$ jest oznaczana minimalna długość β -reguły decyzyjnej dla T i r . Rozważana reguła decyzyjna jest nazywana *zrealizowana dla r* , jeśli $b_1 = \delta_{i_1}, \dots, b_m = \delta_{i_m}$. Rozważana reguła jest *β -prawdziwa dla T* , jeśli d jest najbardziej wspólną decyzją dla T' i $P(T') \leq \beta P(T)$. System S reguł decyzyjnych nad T jest nazywany *β -kompletnym systemem reguł decyzyjnych dla T* , jeśli każda reguła z S jest β -prawdziwa dla T i dla każdego wiersza r z T istnieje reguła w S , która jest zrealizowana dla r . Przez $L(S)$ jest oznaczana maksymalna długość reguły z S , przez $L_\beta(T)$ jest oznaczana minimalna wartość $L(S)$, gdzie minimum jest rozważane spośród wszystkich β -kompletnych systemów reguł decyzyjnych dla T .

Powiązania pomiędzy β -drzewami, β -regułami i β -testami oraz granice ich złożoności

Twierdzenia 1, 2, 3 i Wnioski 1, 2, 3 opisujące relacje pomiędzy drzewami decyzyjnymi, regułami i testami mogą zostać rozszerzone dla przypadku β -drzew decyzyjnych, β -reguł decyzyjnych i β -testów.

Nie można uzyskać nietrywialnej dolnej granicy wartości $h_\beta(T)$ zależnej od wartości $D(T)$, która oznacza liczbę różnych decyzji przypisanych do wierszy tabeli T .

Twierdzenie 18. *Dla dowolnej liczby rzeczywistej β , $0 < \beta < 1$, i dla dowolnej liczby naturalnej m , istnieje tablica decyzyjna T taka, że $D(T) = m + 1$ i $h_\beta(T) = 1$.*

Niech T będzie tablicą decyzyjną z n kolumnami oznaczonymi przez atrybuty f_1, \dots, f_n , i β będzie liczbą rzeczywistą taką, że $0 \leq \beta < 1$. Poniżej zostanie zdefiniowany parametr $M_\beta(T)$ dla tabeli T . Jeśli T jest tabelą zdegenerowaną wówczas $M_\beta(T) = 0$. Niech T będzie tabelą niezdegenerowaną i $\bar{\delta} = (\delta_1, \dots, \delta_n) \in \{0, 1\}^n$. Przez $M_\beta(T, \bar{\delta})$ jest oznaczana minimalna liczba naturalna m taka, że istnieją atrybuty $f_{i_1}, \dots, f_{i_m} \in \{f_1, \dots, f_n\}$ dla których $P(T(f_{i_1}, \delta_{i_1}) \dots (f_{i_m}, \delta_{i_m})) \leq \beta P(T)$. Wówczas $M_\beta(T) = \max\{M_\beta(T, \bar{\delta}) : \bar{\delta} \in \{0, 1\}^n\}$.

Twierdzenia 5, 6, 7 i Wniosek 4 opisujące związki pomiędzy $L(T, r)$, $L(T)$, $M(T)$, $M(T, \bar{\delta})$, $h(T)$ i $R(T)$ mogą zostać rozszerzone dla przypadku $L_\beta(T, r)$, $L_\beta(T)$, $M_\beta(T)$, $M_\beta(T, \bar{\delta})$, $h_\beta(T)$, i $R_\beta(T)$. Istnieją także nowe rodzaje granic, m.in:

Twierdzenie 19. *Niech T będzie niezdegenerowaną tablicą decyzyjną zawierającą n kolumn oznaczonych przez atrybuty f_1, \dots, f_n , $\bar{\delta} = (\delta_1, \dots, \delta_n) \in \{0, 1\}^n$, i β będzie liczbą rzeczywistą taką, że $0 \leq \beta < 1$. Wówczas $M_\beta(T, \bar{\delta}) \leq [(1 - \beta)M(T, \bar{\delta})]$.*

Wniosek 5. $L_\beta(T, r) \leq [(1 - \beta)L(T, r)]$ dla dowolnego wiersza r z T .

Wniosek 6. $L_\beta(T) \leq [(1 - \beta)L(T)]$.

Algorytmy aproksymacyjne dla optymalizacji β -reguł decyzyjnych

Niech T będzie niezdegenerowaną tablicą decyzyjną zawierającą n kolumn oznaczonych przez atrybuty f_1, \dots, f_n , $r = (b_1, \dots, b_n)$ będzie wierszem z T i β będzie liczbą rzeczywistą taką, że $0 < \beta < 1$. Przez $A(T)$ jest oznaczany zbiór wszystkich nieuporządkowanych par wierszy z T o różnych decyzjach. Powiemy, że atrybut f_i separuje parę $(r_1, r_2) \in A(T)$, jeśli wiersze r_1 i r_2 mają różne liczby na przecięciu z kolumną f_i . Oczywiście, $|A(T)| = P(T)$.

Algorytm 3 przedstawia *algorytm zachłanny dla konstruowania β -reguły decyzyjnej* dla T i r . Wynikiem działania algorytmu jest β -reguła decyzyjna dla T i r : $f_{i_1} = b_{i_1} \wedge \dots \wedge f_{i_m} = b_{i_m} \rightarrow d$, gdzie d jest najbardziej wspólną decyzją dla tabeli $T(f_{i_1}, b_1) \dots (f_{i_m}, b_m)$. Przez $L_{\text{greedy}}(T, r, \beta)$ jest oznaczana długość skonstruowanej reguły.

Algorytm 3: Algorytm zachłanny dla konstruowania β -reguły decyzyjnej dla T i r .

Wejście: Tablica decyzyjna T z atrybutami f_1, \dots, f_n , wiersz $r = (b_1, \dots, b_n)$ w T , liczba rzeczywista β , $0 < \beta < 1$.

Wyjście: β -reguła decyzyjna dla T i r .

$Q \leftarrow \emptyset$;

while atrybuty z Q separują mniej niż $\beta|A(T)|$ par wierszy z $A(T)$ **do**

wybierz $f_i \in \{f_1, \dots, f_n\}$ o minimalnym indeksie i taki, że f_i separuje od r maksymalną liczbę par wierszy z $A(T)$ nieodseparowanych przez atrybuty z Q ;

$Q \leftarrow Q \cup \{f_i\}$;

end

return $\bigwedge_{f_i \in Q} (f_i = b_i) \rightarrow d$;

Twierdzenie 20. Niech T będzie niezdegenerowaną tablicą decyzyjną, r będzie wierszem z T i β będzie liczbą rzeczywistą taką, że $0 < \beta < 1$. Wówczas $L_{\text{greedy}}(T, r, \beta) \leq L(T, r) \ln \frac{1}{\beta} + 1$.

Można wykorzystać rozważany algorytm dla skonstruowania β -kompletnego systemu reguł decyzyjnych dla T . Wówczas algorytm ten jest stosowany sekwencyjnie dla tabeli T , liczby β i każdego wiersza r z T . Jako wynik zostaje skonstruowany system reguł S , w którym każda reguła jest β -prawdziwa dla T i dla każdego wiersza z T istnieje reguła w S , która jest zrealizowana dla tego wiersza. $L_{\text{greedy}}(T, \beta) = L(S)$. Z Twierdzenia 20 wynika

Twierdzenie 21. Niech T będzie niezdegenerowaną tablicą decyzyjną i β będzie liczbą rzeczywistą taką, że $0 < \beta < 1$. Wówczas $L_{\text{greedy}}(T, \beta) \leq L(T) \ln \frac{1}{\beta} + 1$.

Poniższe dwa twierdzenia zostały także udowodnione.

Twierdzenie 22. Dla dowolnego β , $0 \leq \beta < 1$, problem minimalizacji długości β -reguły decyzyjnej jest NP-trudny.

Twierdzenie 23. Problem optymalizacji systemu β -reguł decyzyjnych jest NP-trudny dla dowolnego β , $0 \leq \beta < 1$.

Dokładne algorytmy dla optymalizacji β -reguł decyzyjnych

Poniżej zostanie opisany algorytm V_β dla minimalizacji długości β -reguły decyzyjnej.

Pierwsza część pracy algorytmu V_β pokrywa się z pierwszą częścią pracy algorytmu V . Jako wynik zostaje skonstruowany zbiór $SEP(T)$ separowalnych podtabel tabeli T .

Druga część pracy algorytmu V_β dotyczy konstruowania dla każdego wiersza r danej tablicy decyzyjnej T , β -reguły decyzyjnej dla T i r , która posiada minimalną długość. Zaczynając od najmniejszych podtabel ze zbioru $SEP(T)$, algorytm V_β w każdym kroku przypisuje regułę decyzyjną do każdego wiersza r podtabeli $T' \in SEP(T)$.

Przypuśćmy, że $p \geq 0$ kroków drugiej części algorytmu V_β zostało właśnie wykonanych.

Krok ($p + 1$): Jeśli każdy wiersz r tabeli T jest oznaczony przez regułę decyzyjną, wówczas reguła przypisana do r jest wynikiem pracy V_β dla T i r (reguła ta jest oznaczona jako $V_\beta(T, r)$). W przeciwnym przypadku, wybierz ze zbioru $SEP(T)$ tabelę D spełniającą następujące warunki:

- a) wiersze w D nie są oznaczone przez reguły decyzyjne;
- b) $P(D) \leq \beta P(T)$ lub $P(D) > \beta P(T)$ i dla wszystkich separowalnych podtabeli D , rodzaju $D(f_i, \delta)$, $f_i \in E(D)$, $\delta \in \{0, 1\}$, każdy wiersz ma przypisaną regułę decyzyjną.

Niech $P(D) \leq \beta P(T)$ i d będzie najbardziej wspólną decyzją dla D . Wówczas przypisz do każdego wiersza w D regułę decyzyjną $\rightarrow d$.

Niech $P(D) > \beta P(T)$ i $r = (\delta_1, \dots, \delta_n)$ będzie wierszem z D . Dla dowolnego $f_i \in E(D)$, konstruowana jest reguła $rule(r, f_i)$. Niech wiersz r w tabeli $D(f_i, \delta_i)$ ma przypisaną regułę $\beta_i \rightarrow d_i$. Wówczas reguła $rule(r, f_i)$ jest równa $f_i = \delta_i \wedge \beta_i \rightarrow d_i$. Oznacz wiersz r tabeli D przez jedną z reguł $rule(r, f_i)$, $f_i \in E(D)$, posiadającą minimalną długość i przejdź do kroku ($p + 2$).

Twierdzenie 24. Niech β będzie liczbą rzeczywistą taką, że $0 \leq \beta < 1$. Wówczas, dla dowolnej niezdegenerowanej tablicy decyzyjnej T i dowolnego wiersza r w T , algorytm V_β konstruuje β -regułę decyzyjną $V_\beta(T, r)$ dla T i r o minimalnej długości $L_\beta(T, r)$. Podczas konstruowania optymalnych reguł dla wierszy tabeli T algorytm V_β wykonuje $2|SEP(T)| + 3$ kroków. Czas pracy algorytmu V_β jest ograniczony z dołu przez $|SEP(T)|$ i ograniczony z góry przez wielomian od $|SEP(T)|$ i od liczby kolumn w tabeli T .

5.6 Uczenie nadzorowane – klasyfikatory oparte na regułach decyzyjnych

W poprzednich rozdziałach były rozważane algorytmy dla konstruowania systemów reguł decyzyjnych dla danej tablicy decyzyjnej T . Jednym z najbardziej znanych zastosowań takich systemów jest wykorzystanie ich jako klasyfikatorów. Jeśli T zawiera kompletną informację (wszystkie możliwe n -tki wartości atrybutów są wierszami w T) wówczas, zależnie od celu, konstruowane są albo dokładne albo przybliżone klasyfikatory o danej dokładności.

Jeśli T nie zawiera kompletnych informacji (nie wszystkie możliwe n -tki wartości atrybutów i odpowiadające im decyzje znajdują się w tabeli), wówczas problem jest bardziej złożony i znany w literaturze jako *uczenie nadzorowane*. Dla danej tablicy decyzyjnej T z atrybutami warunkowymi f_1, \dots, f_n i atrybutem decyzyjnym d , należy skonstruować klasyfikator, który zapewni predykcję wartości atrybutu decyzyjnego dla n -tek wartości atrybutów warunkowych, które możliwe, że nie są wierszami tabeli T . W tym przypadku dokładne klasyfikatory mogą być przeuczone tzn. mogą posiadać dobrą jakość klasyfikacji dla T i złą dla n -tek wartości atrybutów, które nie są wierszami z T .

W tej sytuacji, zazwyczaj, początkowa tablica T jest dzielona na trzy podtabele: trenującą podtabelę T_1 , walidacyjną podtabelę T_2 i testową podtabelę T_3 . Podtabela T_1 służy do konstruowania klasyfikatora. Podtabela T_2 służy do przycinania tego klasyfikatora: krok po kroku jakość klasyfikatora względem T_1 maleje, w wyniku usuwania warunków (deskryptorów) z lewej strony reguł decyzyjnych. Przycinanie zostaje zakończone, kiedy jakość uzyskanego klasyfikatora względem T_2 będzie maksymalna. Podtabela T_3 jest stosowana, w celu ewaluacji jakości klasyfikatora uzyskanego po przycięciu. Jeśli jakość jest wystarczająco dobra, klasyfikator ten może zostać użyty, w celu predykcji decyzji dla n -tek wartości atrybutów, które nie są wierszami w T .

Wymienione w poprzednich rozdziałach podejścia dla konstruowania kompletnego systemu reguł decyzyjnych lub β -kompletnego systemu reguł decyzyjnych, $0 \leq \beta < 1$, są następujące:

- algorytmy zachłanne (opisane w rozdziale 5.3 oraz w rozdziale 5.5),
- algorytmy dynamicznego programowania (opisane w rozdziale 5.3 oraz w rozdziale 5.5),
- konstruowanie systemu reguł na podstawie testu,

Niech $F = \{f_{i_1}, \dots, f_{i_m}\}$ będzie testem dla T . Dla każdego wiersza r z T , jest konstruowana reguła decyzyjna $f_{i_1} = a_1 \wedge \dots \wedge f_{i_m} = a_m \rightarrow t$, gdzie a_1, \dots, a_m są liczbami na przecięciu wiersza r i kolumn f_{i_1}, \dots, f_{i_m} , t jest decyzją przypisaną do wiersza r . Ponieważ F jest testem dla T , uzyskany system reguł decyzyjnych jest kompletnym system reguł dla T .

Niech β będzie liczbą rzeczywistą, $0 < \beta < 1$, i $F = \{f_{i_1}, \dots, f_{i_m}\}$ będzie β -testem dla T . Dla każdego wiersza r z T , jest konstruowana reguła decyzyjna $f_{i_1} = a_1 \wedge \dots \wedge f_{i_m} = a_m \rightarrow t$, gdzie a_1, \dots, a_m są liczbami na przecięciu wiersza r i kolumn f_{i_1}, \dots, f_{i_m} , t jest najbardziej wspólną decyzją dla tabeli $T(f_{i_1}, a_1) \dots (f_{i_m}, a_m)$. Ponieważ F jest β -testem dla T , uzyskany system reguł decyzyjnych jest β -kompletnym systemem reguł dla T .

- konstruowanie systemu reguł na podstawie drzewa decyzyjnego.

Niech Γ będzie β -drzewem decyzyjnym, $0 \leq \beta < 1$. Jeśli $\beta = 0$ wówczas Γ jest dokładnym drzewem decyzyjnym. Niech τ będzie ścieżką w Γ prowadzącą od korzenia do węzła końcowego. Na tej ścieżce węzły niekońcowe są oznaczone przez atrybuty f_{i_1}, \dots, f_{i_m} , krawędzie są oznaczone przez liczby b_1, \dots, b_m , i węzeł końcowy τ ma przypisaną decyzję t . Reguła decyzyjna $f_{i_1} = b_1 \wedge \dots \wedge f_{i_m} = b_m \rightarrow t$ odpowiada τ . Wiadomo, że zbiór reguł decyzyjnych odpowiadających ścieżkom w Γ prowadzącym od korzenia do węzłów końcowych jest β -kompletnym systemem reguł decyzyjnych dla T . W szczególności, jeśli Γ jest drzewem decyzyjnym dla T , wówczas rozważany system reguł decyzyjnych jest kompletnym systemem reguł dla T (zobacz Twierdzenie 3).

Uproszczenie systemu reguł

Niech $0 \leq \beta < 1$ i S będzie β -kompletnym systemem reguł decyzyjnych dla T skonstruowanym przez jeden z rozważanych algorytmów. Można próbować uprościć system S przez minimalizację całkowitej długości reguł w S i liczby reguł w S . Niech

$$f_{i_1} = a_1 \wedge \dots \wedge f_{i_m} = a_m \rightarrow t \quad (1)$$

będzie regułą z S . Można próbować usunąć z lewej strony reguły (1) tyle warunków $f_{i_j} = a_j$, $j \in \{1, \dots, m\}$, że dla pozostałych warunków $f_{i_{j(1)}} = a_{j(1)}, \dots, f_{i_{j(k)}} = a_{j(k)}$ dla podtabeli $T' = T(f_{i_{j(1)}}, a_{j(1)}) \dots (f_{i_{j(k)}}, a_{j(k)})$, zachodzi nierówność $P(T') \leq \beta P(T)$. Wówczas reguła

$$f_{i_{j(1)}} = a_{j(1)} \wedge \dots \wedge f_{i_{j(k)}} = a_{j(k)} \rightarrow t' \quad (2)$$

jest dodawana do S zamiast reguły (1), gdzie t' jest najbardziej wspólną decyzją dla T' . Oczywiście zbiór reguł uzyskany z S przez zastąpienie zredukowaną regułą (2) każdej reguły (1) z S , jest systemem β -reguł decyzyjnych dla T .

Reguła z S *pokrywa* wiersz r z T jeśli reguła ta jest zrealizowana dla wiersza r . Ponieważ S jest β -kompletnym systemem reguł decyzyjnych dla T , reguły z S pokrywają wszystkie wiersze z T . Niech S' będzie podsystemem S takim, że reguły z S' pokrywają wszystkie wiersze z T . Wiadomo, że S' jest β -kompletnym systemem reguł decyzyjnych dla T . W celu zmniejszenia liczby reguł w S' , można zastosować algorytm zachłanny dla problemu pokrycia zbioru.

System reguł jako klasyfikator

Niech T będzie tablicą decyzyjną z atrybutami warunkowymi f_1, \dots, f_n i atrybutem decyzyjnym d . Niech S będzie β -kompletnym systemem reguł decyzyjnych dla T . System S może zostać użyty jako klasyfikator, w celu predykcji wartości atrybutu decyzyjnego d dla n-tki $\bar{a} = (a_1, \dots, a_n)$ wartości atrybutów warunkowych f_1, \dots, f_n , w przypadku gdy \bar{a} nie jest wierszem z T . Reguła decyzyjna $f_{i_1} = b_1 \wedge \dots \wedge f_{i_m} = b_m \rightarrow t$ jest nazywana *zrealizowaną dla \bar{a}* jeśli $b_1 = a_{i_1}, \dots, b_m = a_{i_m}$. Jeżeli S nie zawiera reguł, które są zrealizowane dla \bar{a} , wówczas wartość atrybutu d dla \bar{a} jest równa najbardziej wspólnej decyzji dla T . Niech S zawiera reguły, które są zrealizowane dla \bar{a} i c niech będzie minimalną wartością d taką, że liczba reguł z S , które są zrealizowane dla \bar{a} i mają c z prawej strony jest maksymalna. Wówczas wartość d dla \bar{a} jest równa c .

Przycinanie

Kompletny system reguł lub β -kompletny system reguł dla małych wartości β , stosowany jako klasyfikator może być przeuczony tzn. może mieć dobrą jakość klasyfikacji dla wierszy z T i niską jakość klasyfikacji dla n-tek, które nie są wierszami w T . W celu poprawy tej sytuacji można zastosować procedurę przycinania reguł decyzyjnych.

Niech niezdegenerowana tablica decyzyjna T będzie podzielona na trzy niezdegenerowane podtabele T_1, T_2 , i T_3 . Jedno z przedstawionych wcześniej podejść

może zostać wykorzystane do konstrukcji β -kompletnego systemu reguł decyzyjnych dla T_1 . Następnie, system ten może zostać uproszczony. Niech S będzie uzyskanym β -kompletnym systemem reguł decyzyjnych dla T_1 . Rozważmy dowolną regułę z S

$$f_{i_1} = a_1 \wedge \dots \wedge f_{i_m} = a_m \rightarrow t. \quad (3)$$

Dla $j \in \{1, \dots, m\}$, badana jest *podreguła*

$$f_{i_j} = a_j \wedge \dots \wedge f_{i_m} = a_m \rightarrow t' \quad (4)$$

reguły (3), gdzie t' jest najbardziej wspólną decyzją dla podtabeli $T_1(f_{i_j}, a_j) \dots (f_{i_m}, a_m)$, i można wyznaczyć *niedokładność* $\frac{P(T_1(f_{i_j}, a_j) \dots (f_{i_m}, a_m))}{P(T_1)}$ podreguły (4) względem tabeli T_1 .

Niech $\beta_1 < \beta_2 < \dots < \beta_q$ będą wszystkimi różnymi niedokładnościami dla wszystkich podreguł reguł z S , które są większe lub równe β . Dla każdego $k \in \{1, \dots, q\}$, konstruowany jest system reguł decyzyjnych S_k . Dla dowolnej reguły (3) z S , podreguła (4) reguły (3) z maksymalnym $j \in \{1, \dots, m\}$ dla którego niedokładność (4) wynosi najwyższej β_k , jest dodawana do S_k .

Dla każdego $k \in \{1, \dots, q\}$, system reguł decyzyjnych S_k jest stosowany jako klasyfikator dla tabeli T_2 i obliczana jest liczba błędnych zaklasyfikowań. Jest to liczba wierszy w T_2 dla których decyzja wyznaczona przez system S_k nie jest równa decyzji dołączonej do rozważanego wiersza.

Wybierane jest minimalne $k_0 \in \{1, \dots, q\}$, dla którego system reguł S_{k_0} posiada minimalną liczbę błędnych zaklasyfikowań. System ten S_{k_0} jest traktowany jako ostateczny klasyfikator. Jest on stosowany do tabeli T_3 i obliczana jest jego jakość klasyfikacji (liczba błędnych zaklasyfikowań dla wierszy z tabeli T_3).

Rozdział ten był poświęcony rozważaniom dotyczącym problemu nadzorowanego uczenia: dla danej tablicy decyzyjnej T i n -tki $\bar{\delta}$ wartości atrybutów warunkowych z T , która nie jest wierszem w T , należy dokonać predykcji wartości atrybutu decyzyjnego dla $\bar{\delta}$. W tym celu konstruowany jest klasyfikator oparty na regułach decyzyjnych. Inną możliwością jest pominięcie konstrukcji klasyfikatorów i wykorzystanie informacji zawartych w T i $\bar{\delta}$ bezpośrednio. W tym przypadku, mogą być stosowane leniwe algorytmy uczenia oparte na regułach, zarówno deterministycznych (decyzyjnych) jak i niedeterministycznych (wzbraniających) [23].

5.7 Lokalne podejście dla badania drzew decyzyjnych i systemów reguł

W tym rozdziale rozważane jest lokalne podejście do badania drzew decyzyjnych i systemów reguł decyzyjnych, dla problemów nad skończonymi i nieskończonymi systemami informacyjnymi. Lokalne podejście opiera się na założeniu, że tylko atrybuty zawarte w opisie problemu, zostaną użyte w drzewach decyzyjnych i systemach reguł decyzyjnych rozwiązujących ten problem. Istnieje także globalne podejście, które opiera się na założeniu, że dowolne atrybuty z rozważanego systemu informacyjnego mogą zostać użyte w drzewach decyzyjnych

i systemach reguł decyzyjnych rozwiązujących dany problem. Jednym z celów tego rozdziału jest porównanie złożoności czasowej drzew decyzyjnych i systemów reguł decyzyjnych.

Podstawowe pojęcia

Niech A będzie niepustym zbiorem, B będzie skończonym, niepustym zbiorem zawierającym przynajmniej dwa elementy i F będzie niepustym zbiorem funkcji odwzorowujących A na B . Funkcje z F są nazywane *atrybutami*, trójka $U = (A, B, F)$ jest nazywana *systemem informacyjnym*. Jeśli F jest skończonym zbiorem, wówczas U jest nazywany *skończonym* systemem informacyjnym. Jeśli F jest nieskończonym zbiorem, wówczas U jest nazywany *nieskończonym* systemem informacyjnym.

Problemem nad U jest dowolna $(n + 1)$ -tka $z = (\nu, f_1, \dots, f_n)$, gdzie $\nu : B^n \rightarrow \omega$, $\omega = \{0, 1, 2, \dots\}$, $f_1, \dots, f_n \in F$. Liczba $\dim z = n$ jest nazywana *wymiarem* problemu z . Problem z może być interpretowany jako problem znalezienia wartości $z(a) = \nu(f_1(a), \dots, f_n(a))$ dla dowolnego $a \in A$. W tej postaci mogą być reprezentowane problemy związane z rozpoznawaniem wzorców, dyskretną optymalizacją, diagnostyką błędów, geometrią obliczeniową. Przez $\mathcal{P}(U)$ jest oznaczany zbiór wszystkich problemów nad systemem informacyjnym U .

Drzewa decyzyjne i systemy reguł decyzyjnych są rozważane jako algorytmy dla rozwiązania danego problemu.

Drzewo decyzyjne nad U jest skończonym drzewem z korzeniem, każdy węzeł końcowy ma przypisaną liczbę z ω , każdy węzeł niekońcowy (węzły takie nazywane są *pracującymi*) jest oznaczony przez atrybut z F , każda krawędź jest oznaczona przez element z B . Krawędzie wychodzące z węzłów niekońcowych są oznaczone przez parami różne elementy z B .

Niech Γ będzie drzewem decyzyjnym nad U . *Kompletna ścieżka* w Γ jest dowolną sekwencją $\xi = v_1, d_1, \dots, v_m, d_m, v_{m+1}$ węzłów i krawędzi w Γ taką, że v_1 jest korzeniem, v_{m+1} jest węzłem końcowym oraz v_i jest początkowym, v_{i+1} jest końcowym węzłem krawędzi d_i , dla $i = 1, \dots, m$. Teraz, zostanie zdefiniowany system równań $\mathcal{S}(\xi)$ i podzbiór $\mathcal{A}(\xi)$ zbioru A związany z ξ . Jeśli $m = 0$ wówczas $\mathcal{S}(\xi)$ jest pustym systemem i $\mathcal{A}(\xi) = A$. Niech $m > 0$, węzeł v_i będzie oznaczony przez atrybut f_i , krawędź d_i będzie oznaczona przez element δ_i z B , $i = 1, \dots, m$. Wówczas $\mathcal{S}(\xi) = \{f_1(x) = \delta_1, \dots, f_m(x) = \delta_m\}$ i $\mathcal{A}(\xi)$ jest zbiorem rozwiązań $\mathcal{S}(\xi)$ z A .

Drzewo decyzyjne Γ nad U *rozwiązuje* problem z nad U jeśli, dla dowolnego $a \in A$, istnieje kompletna ścieżka ξ w Γ taka, że $a \in \mathcal{A}(\xi)$ i węzeł końcowy w ścieżce ξ ma przypisaną liczbę $z(a)$.

Dla drzew decyzyjnych, jako miara złożoności czasowej, rozważana jest *głębokość* drzewa. Jest to maksymalna liczba niekońcowych węzłów w kompletnej ścieżce w drzewie. Głębokość drzewa decyzyjnego Γ jest oznaczana przez $h(\Gamma)$.

Regułą decyzyjną nad U jest dowolne wyrażenie postaci $f_1 = \delta_1 \wedge \dots \wedge f_m = \delta_m \rightarrow \sigma$, gdzie $f_1, \dots, f_m \in F$, $\delta_1, \dots, \delta_m \in B$ i $\sigma \in \omega$. Reguła ta jest oznaczana przez ϱ . Liczba m jest *dlugością* reguły ϱ . Teraz, zostanie zdefiniowany system równań $\mathcal{S}(\varrho)$ i podzbiór $\mathcal{A}(\varrho)$ zbioru A związany z ϱ . Jeśli $m = 0$ wówczas $\mathcal{S}(\varrho)$

jest pustym systemem i $\mathcal{A}(\varrho) = A$. Niech $m > 0$. Wówczas $S(\varrho) = \{f_1(x) = \delta_1, \dots, f_m(x) = \delta_m\}$ i $\mathcal{A}(\varrho)$ jest zbiorem rozwiązań $S(\varrho)$ z A . Liczba σ jest nazywana *prawą stroną reguły* ϱ .

System reguł decyzyjnych nad U jest niepustym, skończonym zbiorem reguł decyzyjnych nad U . Niech S będzie systemem reguł decyzyjnych nad U i z będzie problemem nad U . System reguł decyzyjnych S jest nazywany *kompletny dla problemu* z jeśli, dla dowolnego $a \in A$, istnieje reguła $\varrho \in S$ taka, że $a \in \mathcal{A}(\varrho)$ i dla każdej reguły $\varrho \in S$ takiej, że a jest rozwiązaniem z $S(\varrho)$, prawa strona reguły ϱ jest równa liczbie $z(a)$.

Dla systemów reguł decyzyjnych, jako miara złożoności czasowej, rozważana jest maksymalna długość $L(S)$ reguły z systemu S . Miara $L(S)$ może być określana jako *głębokość* systemu reguł decyzyjnych S .

Badanie drzew decyzyjnych i systemów reguł decyzyjnych dla problemu $z = (\nu, f_1, \dots, f_n)$, w sytuacji kiedy drzewa i reguły używają tylko atrybutów ze zbioru $\{f_1, \dots, f_n\}$, opiera się na badaniu *tablicy decyzyjnej* $T(z)$ związanej z problemem z . Tabela $T(z)$ posiada n kolumn, które zawierają elementy z B . Wiersz $(\delta_1, \dots, \delta_n)$ jest zawarty w tabeli $T(z)$ wtedy i tylko wtedy, jeśli system równań $\{f_1(x) = \delta_1, \dots, f_n(x) = \delta_n\}$ jest kompatybilny z A (posiada rozwiązanie ze zbioru A). Wiersz ten jest oznaczony przez liczbę $\nu(\delta_1, \dots, \delta_n)$. Dla $i = 1, \dots, n$, i -ta kolumna jest oznaczona przez atrybut f_i . Wiadomo, że drzewo decyzyjne nad $T(z)$ rozwiązuje problem z wtedy i tylko wtedy, jeśli drzewo to jest drzewem decyzyjnym dla $T(z)$. Wiadomo także, że system reguł decyzyjnych nad $T(z)$ jest kompletny dla problemu z wtedy i tylko wtedy, jeśli system ten jest kompletnym systemem reguł decyzyjnych dla $T(z)$.

Jeśli rozważane są dodatkowe atrybuty f_{n+1}, \dots, f_{n+m} , wówczas można badać nowy problem $z' = (\mu, f_1, \dots, f_n, f_{n+1}, \dots, f_{n+m})$ taki, że $\mu(x_1, \dots, x_{n+m}) = \nu(x_1, \dots, x_n)$, i odpowiadającą mu tablicę decyzyjną $T(z')$.

Lokalne podejście do badania drzew decyzyjnych i systemów reguł

Niech $U = (A, B, F)$ będzie systemem informacyjnym. Dla problemu $z = (\nu, f_1, \dots, f_n)$ nad U , przez $h_U^l(z)$ jest oznaczana minimalna głębokość drzewa decyzyjnego nad U , które rozwiązuje problem z i wykorzystuje tylko atrybuty ze zbioru $\{f_1, \dots, f_n\}$. Przez $L_U^l(z)$ jest oznaczana minimalna głębokość kompletnego systemu reguł decyzyjnych dla z , system ten wykorzystuje tylko atrybuty ze zbioru $\{f_1, \dots, f_n\}$. W tej części rozdziału zostaną przedstawione powiązania pomiędzy parametrami $h_U^l(z)$, $L_U^l(z)$ i $\dim z$. W związku z tym, funkcje $h_U^l : \omega \setminus \{0\} \rightarrow \omega$ i $L_U^l : \omega \setminus \{0\} \rightarrow \omega$, są definiowane następująco:

$$h_U^l(n) = \max\{h_U^l(z) : z \in \mathcal{P}(U), \dim z \leq n\},$$

$$L_U^l(n) = \max\{L_U^l(z) : z \in \mathcal{P}(U), \dim z \leq n\}$$

dla dowolnego $n \in \omega \setminus \{0\}$, gdzie $\mathcal{P}(U)$ jest zbiorem problemów nad U . Funkcje h_U^l i L_U^l są nazywane *lokalnymi funkcjami Shannona* dla systemu informacyjnego U . Korzystając z Wniosku 3 można uzyskać $L_U^l(n) \leq h_U^l(n)$ dla dowolnego $n \in \omega \setminus \{0\}$.

System informacyjny $U = (A, B, F)$ spełnia *warunek redukcji z parametrem* $m \in \omega \setminus \{0\}$, jeśli dla każdego kompatybilnego z A systemu równań $\{f_1(x) = \delta_1, \dots, f_r(x) = \delta_r\}$, gdzie $r \in \omega \setminus \{0\}$, $f_1, \dots, f_r \in F$ i $\delta_1, \dots, \delta_r \in B$, istnieje podsystem tego systemu, który posiada taki sam zbiór rozwiązań i zawiera najwyżej m równań. System informacyjny $U = (A, B, F)$ spełnia *warunek redukcji*, jeśli spełnia warunek redukcji z parametrem m dla pewnego $m \in \omega \setminus \{0\}$.

W poniższym twierdzeniu rozważane są kryteria zachowań lokalnej funkcji Shannona h_U^l .

Twierdzenie 25. ([26]) *Niech U będzie systemem informacyjnym. Wówczas zachodzi:*

- a) *jeśli U jest skończonym systemem informacyjnym wówczas $h_U^l(n) = O(1)$;*
- b) *jeśli U jest nieskończonym systemem informacyjnym, który spełnia warunek redukcji wówczas $h_U^l(n) = \Theta(\log_2 n)$;*
- c) *jeśli U jest nieskończonym systemem informacyjnym, który nie spełnia warunku redukcji wówczas $h_U^l(n) = n$ dla każdego $n \in \omega \setminus \{0\}$.*

Zostało udowodnione, że istnieją tylko dwa typy zachowań lokalnej funkcji Shannona L_U^l .

Twierdzenie 26. *Niech $U = (A, B, F)$ będzie systemem informacyjnym. Wówczas zachodzi:*

- a) *jeśli U jest skończonym systemem informacyjnym wówczas $L_U^l(n) = O(1)$;*
- b) *jeśli U jest nieskończonym systemem informacyjnym, który spełnia warunek redukcji wówczas $L_U^l(n) = O(1)$;*
- c) *jeśli U jest nieskończonym systemem informacyjnym, który nie spełnia warunku redukcji wówczas $L_U^l(n) = n$ dla każdego $n \in \omega \setminus \{0\}$.*

Ograniczone binarne systemy informacyjne

Jeśli $B = \{0, 1\}$ wówczas system informacyjny $U = (A, B, F)$ jest nazywany *binarny*. Binarny system informacyjny $U = (A, \{0, 1\}, F)$ jest nazywany *ograniczony*, jeśli spełnia warunek redukcji. Niech $U = (A, \{0, 1\}, F)$ będzie ograniczonym binarnym systemem informacyjnym, który spełnia warunek redukcji z parametrem $m \in \omega \setminus \{0\}$. Niech $z = (\nu, f_1, \dots, f_n)$ będzie problemem nad U i $T(z)$ będzie tablicą decyzyjną odpowiadającą temu problemowi.

Korzystając z wyników w [32], można pokazać, że

$$|SEP(T(z))| \leq 2^m n^m + 1, \quad (5)$$

gdzie $SEP(T(z))$ jest zbiorem wszystkich separowalnych podtabel tabeli $T(z)$ włącznie z tabelą $T(z)$.

Z (5) oraz z Twierdzeń 16 i 24 wynika, że algorytmy dla dokładnej optymalizacji reguł decyzyjnych i β -reguł decyzyjnych (algorytmy V i V_β) posiadają dla tabel $T(z)$ wielomianową złożoność czasową zależną od $\dim z$. W [32] pokazano, że jeśli U nie jest ograniczonym systemem informacyjnym, wówczas dla dowolnej liczby naturalnej n istnieje problem z nad U taki, że $\dim z \leq n$

i $|SET(T(z))| \geq 2^n$. Wspomniane fakty przyczyniły się do zainteresowania badaniem ograniczonych systemów informacyjnych.

Poniżej będą rozważane ograniczone binarne liniowe systemy informacyjne. Niech P będzie zbiorem wszystkich punktów na płaszczyźnie i l będzie linią prostą (w skrócie linią) na płaszczyźnie. Linia ta dzieli płaszczyznę na dwie otwarte półpłaszczyzny H_1 i H_2 oraz linię l . Dwa atrybuty odpowiadają linii l . Pierwszy atrybut przyjmuje wartość 0 dla punktów z H_1 i wartość 1 dla punktów z H_2 i l . Drugi atrybut przyjmuje wartość 0 dla punktów z H_2 i wartość 1 dla punktów z H_1 i l . Przez \mathcal{L} jest oznaczany zbiór wszystkich atrybutów odpowiadających liniom na płaszczyźnie. Systemy informacyjne typu $(P, \{0, 1\}, F)$, gdzie $F \subseteq \mathcal{L}$, są nazywane *binarnymi liniowymi systemami informacyjnymi na płaszczyźnie*. Poniżej zostaną opisane wszystkie ograniczone binarne liniowe systemy informacyjne na płaszczyźnie.

Niech l będzie linią na płaszczyźnie. Przez $\mathcal{L}(l)$ jest oznaczany zbiór wszystkich atrybutów odpowiadających liniom, które są równoległe do l . Niech p będzie punktem na płaszczyźnie. Przez $\mathcal{L}(p)$ jest oznaczany zbiór wszystkich atrybutów odpowiadających liniom, które przechodzą przez punkt p . Zbiór atrybutów C z \mathcal{L} jest nazywany *klonem* jeśli $C \subseteq \mathcal{L}(l)$ dla pewnej linii l lub $C \subseteq \mathcal{L}(p)$ dla pewnego punktu p . Dowód poniższego twierdzenia znajduje się w [31].

Twierdzenie 27. *Binarny liniowy system informacyjny na płaszczyźnie $(P, \{0, 1\}, F)$ jest ograniczony wtedy i tylko wtedy jeśli F jest sumą skończonej liczby klonów.*

Lokalne funkcje Shannona dla skończonych systemów informacyjnych

Twierdzenia 25 i 26 dają pewną informację odnośnie zachowania lokalnych funkcji Shannona dla nieskończonych systemów informacyjnych. Dla skończonego systemu informacyjnego U , istnieją tylko powiązania $h_U^l(n) = O(1)$ i $L_U^l(n) = O(1)$. Niemniej jednak, skończone systemy informacyjne są ważne z punktu widzenia różnych zastosowań.

Poniżej zostanie zbadane zachowanie lokalnych funkcji Shannona dla dowolnego skończonego systemu informacyjnego $U = (A, B, F)$ takiego, że $f \neq \text{const}$ dla dowolnego $f \in F$.

Zbiór $\{f_1, \dots, f_n\} \subseteq F$ jest nazywany *zbędny* jeśli $n \geq 2$ i istnieje $i \in \{1, \dots, n\}$ i $\mu: B^{n-1} \rightarrow B$ takie, że $f_i(a) = \mu(f_1(a), \dots, f_{i-1}(a), f_{i+1}(a), \dots, f_n(a))$, dla każdego $a \in A$. Jeśli zbiór $\{f_1, \dots, f_n\}$ nie jest zbędny, wówczas jest nazywany *niezbędny*. Przez $ir(U)$ jest oznaczana maksymalna liczba atrybutów w niezbędnym podzbiore zbioru F .

Systemem równań nad U to dowolny system

$$\{f_1(x) = \delta_1, \dots, f_n(x) = \delta_n\} \quad (6)$$

taki, że $n \in \omega \setminus \{0\}$, $f_1, \dots, f_n \in F$ i $\delta_1, \dots, \delta_n \in B$. System (6) jest nazywany *anulowałny* jeśli $n \geq 2$ i istnieje liczba $i \in \{1, \dots, n\}$ taka, że system $\{f_1(x) = \delta_1, \dots, f_{i-1}(x) = \delta_{i-1}, f_{i+1}(x) = \delta_{i+1}, \dots, f_n(x) = \delta_n\}$ posiada ten

sam zbiór rozwiązań jak system (6). Jeśli system (6) nie jest anulowalny wówczas jest nazywany *nieanulowalny*. Przez $\text{un}(U)$ jest oznaczana maksymalna liczba równań w nieanulowalnym kompatybilnym systemie nad U . Można pokazać, że $1 \leq \text{un}(U) \leq \text{ir}(U)$. Twierdzenie 28 wynika bezpośrednio z [28].

Twierdzenie 28. *Niech $U = (A, B, F)$ będzie skończonym systemem informacyjnym takim, że $f \neq \text{const}$ dla dowolnego $f \in F$ i $n \in \omega \setminus \{0\}$. Wówczas zachodzi:*

- a) jeśli $n \leq \text{un}(U)$ wówczas $h_U^l(n) = n$;
- b) jeśli $\text{un}(U) \leq n \leq \text{ir}(U)$ wówczas

$$\max\{\text{un}(U), \log_k(n+1)\} \leq h_U^l(n) \leq \min\{n, 2(\text{un}(U))^2 \log_2(2kn+2)\},$$

gdzie $k = |B|$;

- c) jeśli $n \geq \text{ir}(U)$ wówczas $h_U^l(n) = h_U^l(\text{ir}(U))$.

Twierdzenie 29 zostało uzyskane na podstawie wyników umieszczonych w [28] dla niedeterministycznych drzew decyzyjnych.

Twierdzenie 29. *Niech $U = (A, B, F)$ będzie skończonym systemem informacyjnym takim, że $f \neq \text{const}$ dla dowolnego $f \in F$, $n \in \omega \setminus \{0\}$. Wówczas zachodzi:*

- a) jeśli $n \leq \text{un}(U)$ wówczas $L_U^l(n) = n$;
- b) jeśli $n \geq \text{un}(U)$ wówczas $L_U^l(n) = \text{un}(U)$.

Problem wyznaczenia wartości $\text{un}(U)$ i $\text{ir}(U)$ dla danego skończonego systemu informacyjnego jest skomplikowanym problemem. Jednak uzyskane wyniki pozwalają znacznie ograniczyć klasę możliwych typów lokalnych funkcji Shanon dla skończonych systemów informacyjnych.

Przykład 2. Przez P jest oznaczany zbiór wszystkich punktów na płaszczyźnie. Rozważmy dowolną linię prostą l , która dzieli płaszczyznę na pozytywną i negatywną półpłaszczyznę i linię l . Do linii l przypisana jest funkcja $f : P \rightarrow \{0, 1\}$. Funkcja f przyjmuje wartość 1, jeśli punkt znajduje się na pozytywnej półpłaszczyźnie i wartość 0, jeśli punkt znajduje się na negatywnej półpłaszczyźnie lub na linii l . Przez F jest oznaczany zbiór funkcji, które odpowiadają pewnemu r wzajemnie rozłącznych skończonych klas równoległych linii prostych. Rozważając skończony system informacyjny $U = (P, \{0, 1\}, F)$ można pokazać, że $\text{ir}(U) = |F|$ i $\text{un}(U) \leq 2r$.

Globalne podejście do badania drzew decyzyjnych i systemów reguł, w szczególności, globalne funkcje Shanon dla nieskończonego systemu informacyjnego, były rozważane w pracy [29]. Globalne funkcje Shanon dla binarnych skończonych systemów informacyjnych były badane w pracy [27] - dla drzew decyzyjnych oraz podczas wykładu plenarnego [40] - dla reguł decyzyjnych.

5.8 Drzewa decyzyjne i reguły nad liniowymi systemami informacyjnymi

W tym rozdziale zostaną przedstawione drzewa decyzyjne i systemy reguł decyzyjnych nad liniowymi systemami informacyjnymi oraz ich zastosowanie dla problemów optymalizacji kombinatorycznej.

Każdy problem nad liniowym systemem informacyjnym może zostać przedstawiony w następującej formie. Rozważmy skończoną liczbę hiperpłaszczyzn w przestrzeni \mathbb{R}^m . Hiperpłaszczyzny dzielą daną część przestrzeni $C \subseteq \mathbb{R}^m$ na dziedziny. Dziedziny te są ponumerowane tak, że różne dziedziny mogą mieć ten sam numer. Dla danego punktu $z \in C$, należy rozpoznać numer dziedziny do której ten punkt należy. Drzewa decyzyjne i reguły decyzyjne nad rozważanym systemem informacyjnym używają atrybutów postaci $\text{sign}(\sum_{i=1}^m a_i x_i + a_{m+1})$. Atrybuty te pozwalają rozpoznać pozycję punktu względem hiperpłaszczyzny zdefiniowanej przez równanie $\sum_{i=1}^m a_i x_i + a_{m+1} = 0$. Przez $\text{Lin}(m)$ jest oznaczany zbiór wszystkich takich atrybutów (atrybuty te są nazywane *liniowymi*). Wynikiem jest liniowy system informacyjny $(C, \{-1, 0, +1\}, \text{Lin}(m))$. Badany jest specjalny rodzaj liniowych systemów informacyjnych $U_p^m = (C_p^m, \{-1, 0, +1\}, \text{Lin}(m))$, gdzie p jest pozytywną liczbą rzeczywistą i C_p^m jest zbiorem rozwiązań w \mathbb{R}^m systemu $\{-p < x_1 < p, \dots, -p < x_m < p\}$.

Górne granice głębokości drzew decyzyjnych i kompletnych systemów reguł decyzyjnych nad U_p^m są rozważane w poniższym twierdzeniu.

Twierdzenie 30. *Niech $m \geq 2$ i $p > 0$. Wówczas dla dowolnego problemu $z \in \mathcal{P}(U_p^m)$ z $\dim z \geq 2$, istnieje:*

- a) *drzewo decyzyjne Γ nad U_p^m , które rozwiązuje problem z i dla którego $h(\Gamma) \leq 2(m+1)^3 \log_2(\dim z + 2m)$;*
- b) *system reguł decyzyjnych S nad U_p^m , który jest kompletny dla z i dla którego $L(S) \leq m+1$.*

Poniżej został przedstawiony przykład problemu optymalizacji uwzględniając następujące ograniczenia: wejścia należą do zbioru C_p^m dla pewnego naturalnego m i pozytywnej liczby rzeczywistej p .

Przykład 3. Problem komiwojażera (n -City traveling salesman problem.) Niech n będzie liczbą naturalną, $n \geq 4$, G_n będzie kompletnym nieskierowanym grafem z n wierzchołkami. Niech każda krawędź G_n będzie oznaczona przez liczbę rzeczywistą a_l taką, że $-p < a_l < p$. Liczba ta jest interpretowana jako długość krawędzi e . Należy znaleźć cykl Hamiltona w G_n o minimalnej długości. Problem ten może być przedstawiony jako problem z_n nad liniowym systemem informacyjnym $U_p^m = (C_p^m, \{-1, 0, +1\}, \text{Lin}(m))$, gdzie $m = (n(n-1))/2$ i $\dim z_n \leq ((n-1)!/2)^2$. Z Twierdzenia 30 wynika, że istnieje drzewo decyzyjne Γ_n nad U_p^m , które rozwiązuje z_n i dla którego $h(\Gamma_n) \leq 4n^7 \log_2 n$. Z Twierdzenia 30 wynika, że istnieje system reguł decyzyjnych S nad U_p^m , który jest kompletny dla z_n i dla którego $L(S_n) \leq (n(n-1)/2) + 1$.

6 Rozszerzenia podejścia opartego na dynamicznym programowaniu dla analizy i optymalizacji reguł decyzyjnych

Rozdział ten przedstawia wyniki badań na podstawie czterech artykułów [9, 10, 65, 70]. Opisana tematyka dotyczy podejścia opartego na dynamicznym programowaniu dla optymalizacji reguł decyzyjnych. Rozdział ten zawiera trzy części. W części pierwszej (rozdział 6.1) przedstawione zostały algorytmy dla sekwencyjnej optymalizacji reguł decyzyjnych, m.in. względem długości i pokrycia. W części drugiej (rozdział 6.2) przedstawiona została modyfikacja opisanych wcześniej algorytmów. Zaproponowany algorytm jest mniej złożony i pozwala konstruować reguły, których pokrycie jest bliskie wartościom maksymalnym. Trzecia część (rozdział 6.3) poświęcona jest analizie powiązań pomiędzy pokryciem a długością reguł i systemów reguł.

6.1 Podejście oparte na dynamicznym programowaniu dla sekwencyjnej optymalizacji reguł decyzyjnych

Rozdział ten poświęcony jest badaniu rozszerzeń podejścia opartego na dynamicznym programowaniu, które pozwalają na sekwencyjną optymalizację przybliżonych reguł decyzyjnych, m.in. względem długości i pokrycia, na podstawie pracy [9]. W artykule [70] rozważana jest sekwencyjna optymalizacja względem kombinacji funkcji kosztu: długość, pokrycie, liczba błędnych zaklasyfikowań.

Aby pracować z regułami przybliżonymi została użyta miara niepewności $P(T)$, która jest liczbą nieuporządkowanych par wierszy o różnych decyzjach w tablicy decyzyjnej T . Dla nieujemnej liczby rzeczywistej β , badane są β -reguły decyzyjne, które lokalizują wiersze w podtabelach tabeli T , dla których miara niepewności wynosi najwyżej β . Zaproponowany algorytm konstruuje skierowany graf acykliczny $\Delta_\beta(T)$, którego wierzchołkami są podtabele tablicy decyzyjnej T opisane przez system równań postaci "atrybut = wartość". Algorytm kończy podział tabeli na podtabele, kiedy niepewność podtabeli wynosi najwyżej β .

Graf $\Delta_\beta(T)$ pozwala opisać cały zbiór tzw. nieredukowalnych β -reguł decyzyjnych. Można opisać wszystkie nieredukowalne β -reguły decyzyjne o minimalnej długości, a następnie wśród nich opisać wszystkie reguły o maksymalnym pokryciu. Można także zmienić porządek optymalizacji. W tym rozdziale zostały także przedstawione wyniki eksperymentów.

Podstawowe pojęcia

Dla wygody czytelnika zostaną powtórzone pewne definicje związane z tablicami decyzyjnymi. Należy zauważyć, że w tym rozdziale zostanie przedstawiona nieznacznie zmodyfikowana definicja β -reguł decyzyjnych.

Tablica decyzyjna T jest tabelą zawierającą n kolumn oznaczonych przez atrybuty warunkowe f_1, \dots, f_n . Wiersze tabeli są wypełnione przez nieujemne liczby całkowite interpretowane jako wartości atrybutów warunkowych. Wiersze tabeli

T są parami różne i do każdego wiersza jest przypisana nieujemna liczba całkowita (decyzja), która jest interpretowana jako wartość atrybutu decyzyjnego.

Przez $N(T)$ jest oznaczana liczba wierszy w tabeli T . Przez $P(T)$ jest oznaczana liczba nieuporządkowanych par wierszy o różnych decyzjach. Wartość ta jest interpretowana jako *niepewność* tabeli T . Tabela T jest *zdegenerowana*, jeśli T jest pusta lub wszystkie wiersze z T są oznaczone przez tę samą decyzję. Oczywiście, w tym przypadku $P(T) = 0$. Minimalna wartość decyzji przypisana do maksymalnej liczby wierszy w T jest nazywana *najbardziej wspólną decyzją dla T* .

Niech $f_{i_1}, \dots, f_{i_m} \in \{f_1, \dots, f_n\}$ i a_1, \dots, a_m będą nieujemnymi liczbami całkowitymi. Przez $T(f_{i_1}, a_1) \dots (f_{i_m}, a_m)$ jest oznaczana podtabela tabeli T , która zawiera tylko takie wiersze z T , które posiadają liczby a_1, \dots, a_m na przecięciu z kolumnami f_{i_1}, \dots, f_{i_m} . Takie podtabele (włączając tabelę T) są nazywane *separowalnymi podtabelami* tabeli T .

Przez $E(T)$ jest oznaczany zbiór atrybutów z $\{f_1, \dots, f_n\}$, które nie są stałe w T (posiadają przynajmniej dwie różne wartości). Dla dowolnego $f_i \in E(T)$, przez $E(T, f_i)$ jest oznaczany zbiór wartości atrybutu f_i w T .

Wyrażenie

$$f_{i_1} = a_1 \wedge \dots \wedge f_{i_m} = a_m \rightarrow d \quad (7)$$

jest nazywane *regułą decyzyjną nad T* , jeśli $f_{i_1}, \dots, f_{i_m} \in \{f_1, \dots, f_n\}$, a_1, \dots, a_m, d są nieujemnymi liczbami całkowitymi. Możliwe jest, że $m = 0$, wówczas reguła (7) jest równa $\rightarrow d$.

Niech $r = (b_1, \dots, b_n)$ będzie wierszem tabeli T . Reguła (7) jest *zrealizowana dla r* , jeśli $a_1 = b_{i_1}, \dots, a_m = b_{i_m}$. Jeśli $m = 0$ wówczas reguła (7) jest zrealizowana dla dowolnego wiersza z T .

Niech β będzie nieujemną liczbą rzeczywistą. Reguła (7) jest *β -prawdziwa dla T* , jeśli d jest najbardziej wspólną decyzją dla $T' = T(f_{i_1}, a_1) \dots (f_{i_m}, a_m)$ i $P(T') \leq \beta$. Jeśli $m = 0$, wówczas reguła (7) jest β -prawdziwa dla T , jeśli d jest najbardziej wspólną decyzją dla T i $P(T) \leq \beta$.

Jeśli reguła (7) jest β -prawdziwa dla T i zrealizowana dla r , wówczas reguła (7) jest *β -regułą decyzyjną dla T i r* . Należy zauważyć, jeśli $\beta = 0$ wówczas jest to dokładna reguła decyzyjna dla T i r .

Nieredukowalne β -reguły decyzyjne

Niech (7) będzie regułą decyzyjną nad T . Reguła (7) jest nazywana *nieredukowalną β -regułą decyzyjną dla T i r* jeśli (7) jest β -regułą decyzyjną dla T i r i spełnione są warunki, jeśli $m > 0$:

- (i) $f_{i_j} \in E(T)$, i jeśli $m > 1$ to $f_{i_j} \in E(T(f_{i_1}, a_1) \dots (f_{i_{j-1}}, a_{j-1}))$ dla $j = 2, \dots, m$;
- (ii) jeśli $m = 1$ to $P(T) > \beta$ i jeśli $m > 1$ to $P(T(f_{i_1}, a_1) \dots (f_{i_{m-1}}, a_{m-1})) > \beta$.

Niech τ będzie regułą decyzyjną nad T i τ będzie równe (7). Liczba m deskryptorów po lewej stronie reguły τ jest nazywana *długością* reguły i jest oznaczana jako $l(\tau)$. Jeśli $m = 0$, wówczas długość reguły τ jest równa 0. *Pokrycie* reguły τ to liczba wierszy w tabeli T , dla których τ jest zrealizowana i które mają

przypisaną decyzję d . Pokrycie reguły τ jest oznaczane przez $c(\tau)$. Jeśli $m = 0$, wówczas pokrycie reguły decyzyjnej τ jest równe liczbie wierszy w T , które mają przypisaną decyzję d . Liczba błędnych zaklasyfikowań (pomyłek) reguły τ to liczba wierszy w tabeli T , dla których τ jest zrealizowana i które mają przypisaną decyzję inną niż d . Liczba pomyłek reguły τ jest oznaczana przez $\mu(\tau)$. Jeśli $m = 0$, wówczas liczba pomyłek reguły decyzyjnej τ jest równa liczbie wierszy w T , które mają przypisaną decyzję inną niż d .

Twierdzenie 31. *Niech T będzie niepustą tablicą decyzyjną, r będzie wierszem tabeli T i τ będzie β -regulą decyzyjną dla T i r , która nie jest nieredukowalna. Wówczas w wyniku usunięcia pewnych deskryptorów z lewej strony reguły τ i zmiany decyzji występującej po prawej stronie reguły τ , można uzyskać nieredukowalną β -regulę decyzyjną $\text{irr}(\tau)$ dla T i r taką, że $l(\text{irr}(\tau)) \leq l(\tau)$ i $c(\text{irr}(\tau)) \geq c(\tau)$.*

Niestety, nie można uzyskać podobnych wyników dla liczby błędnych zaklasyfikowań.

Skierowany graf acykliczny $\Delta_\beta(T)$

Poniżej został przedstawiony algorytm, który konstruuje graf $\Delta_\beta(T)$. Na podstawie tego grafu można opisać zbiór nieredukowalnych β -reguł decyzyjnych dla T i dla każdego wiersza r w T . Wierzchołkami grafu są separowalne podtabele tabeli T (zobacz Algorytm 4).

Graf $\Delta_\beta(T)$ jest skierowanym grafem acyklicznym. Wierzchołek grafu jest nazywany *końcowy*, jeśli nie wychodzą z niego żadne krawędzie. Należy zauważyć, że wierzchołek Θ jest końcowy w grafie $\Delta_\beta(T)$ wtedy i tylko wtedy, jeśli $P(\Theta) \leq \beta$.

W dalszej części rozdziału zostaną opisane procedury optymalizacji grafu $\Delta_\beta(T)$. Wynikiem pracy procedury optymalizacji jest graf Γ , który zawiera ten sam zbiór wierzchołków i krawędzi jak graf $\Delta_\beta(T)$. Różnicą jest to, że dowolny wiersz r w każdym niekończącym wierzchołku Θ w grafie Γ ma przypisany niepusty zbiór atrybutów $E_\Gamma(\Theta, r) \subseteq E(\Theta)$, który może się różnić od $E(\Theta)$. Jest także możliwe, że $\Gamma = \Delta_\beta(T)$.

Poniżej, dla każdego wierzchołka Θ w grafie Γ i dla każdego wiersza r wierzchołka Θ , zostanie opisany zbiór reguł $Rul_\Gamma(\Theta, r)$. W tym celu należy przechodzić w grafie Γ zaczynając od końcowych wierzchołków grafu, do wierzchołka T .

Niech Θ będzie końcowym wierzchołkiem w grafie Γ i d będzie najbardziej wspólną decyzją dla Θ . Wówczas $Rul_\Gamma(\Theta, r) = \{\rightarrow d\}$.

Niech teraz Θ będzie niekończącym wierzchołkiem w grafie Γ takim, że dla każdego potomka Θ' wierzchołka Θ i dla każdego wiersza r' tabeli Θ' , zbiór reguł $Rul_\Gamma(\Theta', r')$ został zdefiniowany. Niech $r = (b_1, \dots, b_n)$ będzie wierszem tabeli Θ . Dla dowolnego atrybutu $f_i \in E_\Gamma(\Theta, r)$, jest definiowany zbiór reguł $Rul_\Gamma(\Theta, r, f_i)$: $Rul_\Gamma(\Theta, r, f_i) = \{f_i = b_i \wedge \gamma \rightarrow s : \gamma \rightarrow s \in Rul_\Gamma(\Theta(f_i, b_i), r)\}$. Wówczas $Rul_\Gamma(\Theta, r) = \bigcup_{f_i \in E_\Gamma(\Theta, r)} Rul_\Gamma(\Theta, r, f_i)$.

Twierdzenie 32. *Dla każdego wierzchołka Θ w grafie $\Delta_\beta(T)$ i dla każdego wiersza r tabeli Θ , zbiór $Rul_{\Delta_\beta(T)}(\Theta, r)$ jest równy zbiorowi wszystkich nieredukowalnych β -reguł decyzyjnych dla Θ i r .*

Algorytm 4: Algorytm dla konstruowania grafu $\Delta_\beta(T)$

Wejście: Tablica decyzyjna T zawierająca atrybuty f_1, \dots, f_n , nieujemna liczba rzeczywista β .

Wyjście: Graf $\Delta_\beta(T)$.

Skonstruuj graf zawierający pojedynczy wierzchołek T , który nie jest oznaczony jako “przetworzony”;

```
while wszystkie wierzchołki grafu nie są oznaczone jako “przetworzone” do
  Wybierz wierzchołek (tabelę)  $\Theta$ , który nie został jeszcze “przetworzony”;
  if  $P(\Theta) \leq \beta$  then
    | Badany wierzchołek jest oznaczany jako “przetworzony”;
  end
  if  $P(\Theta) > \beta$  then
    Dla każdego atrybutu  $f_i \in E(\Theta)$ , rysowana jest wiązka krawędzi
    wychodzących z wierzchołka  $\Theta$ .
    Niech  $E(\Theta, f_i) = \{b_1, \dots, b_t\}$ , wówczas rysowanych jest  $t$  krawędzi
    wychodzących z wierzchołka  $\Theta$ . Krawędzie te są oznaczane odpowiednio
    parami  $(f_i, b_1), \dots, (f_i, b_t)$ . Krawędzie te wchodzą do wierzchołków
     $\Theta(f_i, b_1), \dots, \Theta(f_i, b_t)$ . Jeśli jakieś wierzchołki  $\Theta(f_i, b_1), \dots, \Theta(f_i, b_t)$  nie
    istnieją w grafie, wówczas są dodawane do grafu. Każdemu wierszowi  $r$ 
    tabeli  $\Theta$  zostaje przypisany zbiór atrybutów  $E_{\Delta_\beta(T)}(\Theta, r) = E(\Theta)$ .
    Wierzchołek  $\Theta$  jest oznaczany jako “przetworzony”;
  end
end
return Graf  $\Delta_\beta(T)$ ;
```

Procedura optymalizacji względem długości l

Wykonując procedurę optymalizacji grafu Γ względem długości należy przechodzić od końcowych wierzchołków grafu, do wierzchołka T . Każdy wiersz r każdej tabeli Θ będzie miał przypisaną liczbę $Opt_\Gamma^l(\Theta, r)$, która oznacza minimalną długość β -reguły decyzyjnej ze zbioru $Rul_\Gamma(\Theta, r)$ i zostanie zmieniony zbiór $E_\Gamma(\Theta, r)$ przypisany do wiersza r w Θ , jeśli Θ jest niekończącym wierzchołkiem w grafie Γ . Uzyskany graf jest oznaczany przez Γ^l .

Niech Θ będzie końcowym wierzchołkiem w grafie Γ . Wówczas, do każdego wiersza r tabeli Θ jest przypisywana liczba $Opt_\Gamma^l(\Theta, r) = 0$.

Niech Θ będzie niekończącym wierzchołkiem w grafie Γ i wszyscy potomkowie Θ zostali przetworzeni. Niech $r = (b_1, \dots, b_n)$ będzie wierszem tabeli Θ . Do wiersza r w tabeli Θ jest przypisywana liczba $Opt_\Gamma^l(\Theta, r) = \min\{Opt_\Gamma^l(\Theta(f_i, b_i), r) + 1 : f_i \in E_\Gamma(\Theta, r)\}$ i zbiór $E_{\Gamma^l}(\Theta, r) = \{f_i : f_i \in E_\Gamma(\Theta, r), Opt_\Gamma^l(\Theta(f_i, b_i), r) + 1 = Opt_\Gamma^l(\Theta, r)\}$.

Twierdzenie 33. Dla każdego wierzchołka Θ w grafie Γ^l i dla każdego wiersza r w Θ , zbiór $Rul_{\Gamma^l}(\Theta, r)$ jest równy zbiorowi wszystkich β -reguł decyzyjnych o minimalnej długości ze zbioru $Rul_\Gamma(\Theta, r)$.

Procedura optymalizacji względem pokrycia c

Wykonując procedurę optymalizacji grafu Γ względem pokrycia należy przechodzić od końcowych wierzchołków grafu, do wierzchołka T . Każdy wiersz r każdej tabeli Θ będzie miał przypisaną liczbę $Opt_{\Gamma}^c(\Theta, r)$, która oznacza maksymalne pokrycie β -reguły decyzyjnej ze zbioru $Rul_{\Gamma}(\Theta, r)$ i zostanie zmieniony zbiór $E_{\Gamma}(\Theta, r)$ przypisany do wiersza r w Θ , jeśli Θ jest niekończącym wierzchołkiem w grafie Γ . Uzyskany graf jest oznaczany przez Γ^c .

Niech Θ będzie końcowym wierzchołkiem w grafie Γ i d będzie najbardziej wspólną decyzją dla Θ . Wówczas, do każdego wiersza r tabeli Θ jest przypisywana liczba $Opt_{\Gamma}^c(\Theta, r)$, która jest równa liczbie wierszy w Θ , które mają przypisaną decyzję d .

Niech Θ będzie niekończącym wierzchołkiem w grafie Γ i wszyscy potomkowie Θ zostali przetworzeni. Niech $r = (b_1, \dots, b_n)$ będzie wierszem tabeli Θ . Do wiersza r w tabeli Θ jest przypisywana liczba $Opt_{\Gamma}^c(\Theta, r) = \max\{Opt_{\Gamma}^c(\Theta(f_i, b_i), r) : f_i \in E_{\Gamma}(\Theta, r)\}$ i zbiór $E_{\Gamma^c}(\Theta, r) = \{f_i : f_i \in E_{\Gamma}(\Theta, r), Opt_{\Gamma}^c(\Theta(f_i, b_i), r) = Opt_{\Gamma}^c(\Theta, r)\}$.

Twierdzenie 34. *Dla każdego wierzchołka Θ w grafie Γ^c i dla każdego wiersza r w Θ , zbiór $Rul_{\Gamma^c}(\Theta, r)$ jest równy zbiorowi wszystkich β -reguł decyzyjnych o maksymalnym pokryciu ze zbioru $Rul_{\Gamma}(\Theta, r)$.*

Procedura optymalizacji względem liczby błędnych zaklasyfikowań μ

Wykonując procedurę optymalizacji grafu Γ względem liczby błędnych zaklasyfikowań należy przechodzić od końcowych wierzchołków grafu, do wierzchołka T . Każdy wiersz r każdej tabeli Θ będzie miał przypisaną liczbę $Opt_{\Gamma}^{\mu}(\Theta, r)$, która oznacza minimalną liczbę błędnych zaklasyfikowań β -reguły decyzyjnej ze zbioru $Rul_{\Gamma}(\Theta, r)$ i zostanie zmieniony zbiór $E_{\Gamma}(\Theta, r)$ przypisany do wiersza r w Θ , jeśli Θ jest niekończącym wierzchołkiem w grafie Γ . Uzyskany graf jest oznaczany przez Γ^{μ} .

Niech Θ będzie końcowym wierzchołkiem w grafie Γ i d będzie najbardziej wspólną decyzją dla Θ . Wówczas, do każdego wiersza r tabeli Θ jest przypisywana liczba $Opt_{\Gamma}^{\mu}(\Theta, r)$, która jest równa liczbie wierszy w Θ , które mają przypisaną decyzję inną niż d .

Niech Θ będzie niekończącym wierzchołkiem w grafie Γ i wszyscy potomkowie Θ zostali przetworzeni. Niech $r = (b_1, \dots, b_n)$ będzie wierszem tabeli Θ . Do wiersza r w tabeli Θ jest przypisywana liczba $Opt_{\Gamma}^{\mu}(\Theta, r) = \min\{Opt_{\Gamma}^{\mu}(\Theta(f_i, b_i), r) : f_i \in E_{\Gamma}(\Theta, r)\}$ i zbiór $E_{\Gamma^{\mu}}(\Theta, r) = \{f_i : f_i \in E_{\Gamma}(\Theta, r), Opt_{\Gamma}^{\mu}(\Theta(f_i, b_i), r) = Opt_{\Gamma}^{\mu}(\Theta, r)\}$.

Twierdzenie 35. *Dla każdego wierzchołka Θ w grafie Γ^{μ} i dla każdego wiersza r w Θ , zbiór $Rul_{\Gamma^{\mu}}(\Theta, r)$ jest równy zbiorowi wszystkich β -reguł decyzyjnych o minimalnej liczbie błędnych zaklasyfikowań ze zbioru $Rul_{\Gamma}(\Theta, r)$.*

Sekwencyjna optymalizacja i całkowicie optymalne reguły

Twierdzenia 33, 34 i 35 pokazują, że można wykonać sekwencyjną optymalizację względem długości l , pokrycia c i liczby błędnych zaklasyfikowań μ . Można znaleźć wszystkie nieredukowalne β -reguły decyzyjne o minimalnej długości, a następnie między nimi można znaleźć wszystkie reguły o maksymalnym pokryciu, a między nimi można znaleźć wszystkie reguły o minimalnej liczbie błędnych zaklasyfikowań. Można także zmienić porządek optymalizacji i badać dowolny zbiór tych funkcji kosztu.

Niech F będzie jednym z tych zbiorów $\{l, c, \mu\}$, $\{l, c, \}$, $\{l, \mu\}$ i $\{c, \mu\}$. Nieredukowalna β -reguła decyzyjna τ dla T i r jest nazywana *całkowicie optymalna względem funkcji kosztu z F* jeśli dla każdej funkcji kosztu $f \in F$, wartość $f(\tau)$ jest minimalna jeśli $f \in \{l, \mu\}$ lub maksymalna jeśli $f = c$, spośród wszystkich nieredukowalnych β -reguł decyzyjnych dla T i r . Można opisać wszystkie całkowicie optymalne reguły względem funkcji kosztu z F korzystając z procedur optymalizacji względem tych funkcji kosztu.

W celu przedstawienia procesu sekwencyjnej optymalizacji zostanie opisany przypadek, gdy $F = \{l, c\}$. Nieredukowalna β -reguła decyzyjna dla T i r jest nazywana *całkowicie optymalna względem długości i pokrycia*, jeśli posiada minimalną długość i maksymalne pokrycie między wszystkimi nieredukowanymi β -regułami decyzyjnymi dla T i r . W dalszej części rozdziału, dla uproszczenia, reguły te będą nazywane całkowicie optymalne. Można opisać wszystkie całkowicie optymalne reguły korzystając z procedur optymalizacji względem długości i pokrycia.

Niech $\Gamma = \Delta_\beta(T)$. Stosowana jest procedura optymalizacji grafu Γ względem pokrycia. Wynikiem jest graf Γ^c i każdy wiersz r w T , ma przypisaną wartość $Opt_{\Gamma^c}^c(T, r)$, która jest równa maksymalnemu pokryciu nieredukowalnej β -reguły decyzyjnej dla T i r . Następnie zostanie zastosowana procedura optymalizacji grafu Γ względem długości. Wynikiem jest graf Γ^l . Dalej, zostanie zastosowana procedura optymalizacji grafu Γ^l względem pokrycia. Wynikiem jest graf Γ^{lc} i każdy wiersz r w T , ma przypisaną wartość $Opt_{\Gamma^{lc}}^c(T, r)$, która jest równa maksymalnemu pokryciu nieredukowalnej β -reguły decyzyjnej dla T i r między wszystkimi nieredukowanymi β -regułami decyzyjnymi dla T i r o minimalnej długości. Można pokazać, że całkowicie optymalna nieredukowalna β -reguła decyzyjna dla T i r istnieje wtedy i tylko wtedy, gdy $Opt_{\Gamma^c}^c(T, r) = Opt_{\Gamma^{lc}}^c(T, r)$. Jeśli równanie to jest spełnione wówczas zbiór $Rul_{\Gamma^{lc}}(T, r)$ jest równy zbiorowi wszystkich całkowicie optymalnych nieredukowalnych β -reguł decyzyjnych dla T i r .

Sekwencyjna optymalizacja względem długości, pokrycia i liczby błędnych zaklasyfikowań, została opisana w pracy [70]. Znajdują się tam także wyniki eksperymentalne dotyczące całkowicie optymalnych reguł względem pokrycia, długości i liczby błędnych zaklasyfikowań. W celu zdefiniowania przybliżonych reguł decyzyjnych zastosowano inną miarę niepewności dla tablicy decyzyjnej T , która jest różnicą pomiędzy liczbą wierszy w tablicy i liczbą wierszy z najbardziej wspólną decyzją dla tablicy decyzyjnej. Konstruowane reguły zostały nazwane nieredukowanymi γ -regułami decyzyjnymi.

Wyniki eksperymentów

Pełny zestaw wykonanych eksperymentów można zobaczyć w pracach [9, 70]. Zawarte są tam m.in. wyniki dotyczące minimalnej długości i maksymalnego pokrycia przybliżonych reguł decyzyjnych konstruowanych za pomocą algorytmów dynamicznego programowania, wyniki dotyczące sekwencyjnej optymalizacji względem kombinacji funkcji kosztu: długość, pokrycie i liczba błędnych zaklasyfikowań oraz wyniki dotyczące reguł całkowicie optymalnych. Badana była także jakość klasyfikatorów opartych na przybliżonych regułach decyzyjnych zoptymalizowanych względem pokrycia lub długości oraz zoptymalizowanych sekwencyjnie względem długości i pokrycia (zobacz Tabela 2).

Do przeprowadzenia eksperymentów zostały wykorzystane tablice decyzyjne z Repozytorium Uczenia Maszynowego [11]. Niech T będzie jedną z tych tablic decyzyjnych, $\beta = P(T) \times 0.01$.

Tabela 1 przedstawia wyniki dotyczące całkowicie optymalnych nieredukowalnych β -reguł decyzyjnych względem długości i pokrycia. Dla każdego wiersza r z T , została znaleziona liczba całkowicie optymalnych nieredukowalnych β -reguł decyzyjnych dla T i r . Następnie, została znaleziona dla wierszy z T minimalna liczba całkowicie optymalnych reguł (kolumna “min”), maksymalna liczba takich reguł (kolumna “max”) i średnia liczba całkowicie optymalnych reguł (kolumna “avg”). Kolumna “t-opt” zawiera liczbę wierszy r dla których istnieją całkowicie optymalne nieredukowalne β -reguły decyzyjne dla T i r .

Tabela 1: Całkowicie optymalne β -reguły decyzyjne dla $\beta \in \{P(T) \times 0.01\}$

Nazwa tablicy decyzyjnej	Wiersze	$\beta = P(T) \times 0.01$			
		t-opt	min	avg	max
Adult-stretch	16	16	1	1.500	2
Balance-scale	625	625	2	2.739	12
Breast-cancer	266	25	0	0.154	2
Cars	1728	1728	1	1.705	8
Flags	193	37	0	0.238	2
Hayes-roth-data	69	69	1	2.232	6
House-votes-84	279	23	0	0.179	4
Lcnsacs	24	24	1	3.500	6
Lymphography	148	48	0	0.581	2
Mushroom	8124	600	0	0.121	2
Nursery	12960	12960	1	1.667	2
Shuttle-landing	15	13	0	2.600	24
Soybean-small	47	37	0	1.851	5
Teeth	23	23	1	11.391	60
Tic-tac-toe	958	639	0	2.217	12
Zoo	59	38	0	1.000	18

Tabela 2 przedstawia wyniki dotyczące jakości klasyfikatorów regułowych. Jest to średni błąd dwukrotnej walidacji krzyżowej (eksperymenty zostały powtórzone 50 razy dla każdej tablicy decyzyjnej). Każdy zbiór danych został losowo podzielony na trzy części: trenująca–30%, walidacyjna–20% i testowa–50%. Klasyfikator został skonstruowany na podstawie części trenującej. Następnie, reguły zostały przycięte uwzględniając minimalny błąd klasyfikacji dla zbioru walidacyjnego. Dla części trenującej zbioru danych konstruowane były dokładne reguły decyzyjne (0-reguły). Następnie reguły te były przycinane i uzyskano β -reguły decyzyjne dla rosnących wartości β . Została wybrana taka wartość β ,

Tabela 2: Średni błąd klasyfikatorów opartych na regułach decyzyjnych

Nazwa tablicy decyzyjnej	Optymalizacja			
	coverage	coverage & length	length	length & coverage
Balance-scale	0.239	0.236	0.294	0.286
Breast-cancer	0.296	0.299	0.307	0.305
Cars	0.207	0.196	0.217	0.196
Flags	0.441	0.419	0.410	0.434
Hayes-roth-data	0.345	0.338	0.368	0.332
House-votes-84	0.048	0.049	0.082	0.064
Lymphography	0.277	0.254	0.347	0.253
Mushroom	0.001	0.001	0.002	0.001
Nursery	0.053	0.052	0.050	0.052
Soybean-small	0.171	0.097	0.170	0.079
Spect-test	0.081	0.080	0.080	0.080
Zoo	0.180	0.195	0.244	0.233
Average	0.199	0.191	0.217	0.193

dla której uzyskano minimalny błąd klasyfikacji na zbiorze walidacyjnym. Uzyskany zbiór reguł został zastosowany jako klasyfikator dla części testowej i został wyznaczony błąd klasyfikacji. Jest to liczba obiektów (wierszy) z części testowej tablicy decyzyjnej, które nie zostały poprawnie sklasyfikowane, podzielona przez liczbę wszystkich wierszy z części testowej tablicy decyzyjnej. Ostatni wiersz Tabeli 2 przedstawia średni błąd klasyfikacji dla wszystkich tablic decyzyjnych.

Na podstawie przeprowadzonych eksperymentów można zauważyć, że najmniejszy średni błąd klasyfikacji występuje w przypadku sekwencyjnej optymalizacji reguł decyzyjnych względem pokrycia i długości-0.191 oraz w przypadku sekwencyjnej optymalizacji reguł decyzyjnych względem długości i pokrycia-0.193.

6.2 Zmodyfikowany algorytm dynamicznego programowania dla optymalizacji przybliżonych reguł decyzyjnych względem pokrycia

Rozdział ten, na podstawie pracy [65], przedstawia zmodyfikowany algorytm dynamicznego programowania dla konstruowania i optymalizacji reguł decyzyjnych względem pokrycia. Podobnie jak w poprzednim rozdziale, wykorzystana została miara niepewności $P(T)$ i były badane β -reguły decyzyjne, które lokalizują wiersze w podtabelach tabeli T , dla których miara niepewności wynosi najwyżej β . Różnicą jest to, że nie jest konstruowany kompletny skierowany graf acykliczny $\Delta_\beta(T)$ lecz jego część. Zamiast wszystkich niestałych atrybutów i ich wartości, uwzględniany jest tylko jeden atrybut o minimalnej liczbie wartości i wszystkie jego wartości oraz dla pozostałych atrybutów–wybierana jest tylko jedna, najczęstsza wartość danego atrybutu. Rozmiar grafu jest mniejszy niż rozmiar grafu $\Delta_\beta(T)$. Natomiast wartości pokrycia reguł uzyskanych dla tego grafu po zastosowaniu procedury optymalizacji, są zazwyczaj bliskie wartościom pokrycia reguł uzyskanych dla grafu $\Delta_\beta(T)$ po zastosowaniu procedury optymalizacji.

Algorytm dla konstruowania skierowanego grafu acyklicznego

Niech T będzie tablicą decyzyjną z n kolumnami oznaczonymi przez atrybuty warunkowe f_1, \dots, f_n . Przez $E(T)$ jest oznaczany zbiór atrybutów z $\{f_1, \dots, f_n\}$,

Algorytm 5: Algorytm dla konstruowania grafu $\Delta_\beta^*(T)$

Wejście: Tablica decyzyjna T zawierająca atrybuty f_1, \dots, f_n , nieujemna liczba rzeczywista β .

Wyjście: Graf $\Delta_\beta^*(T)$.

Skonstruuj graf zawierający pojedynczy wierzchołek T , który nie jest oznaczony jako “przetworzony”;

while wszystkie wierzchołki grafu nie są oznaczone jako “przetworzone” **do**

 Wybierz wierzchołek (tabelę) Θ , który nie został jeszcze “przetworzony”;

if $P(\Theta) \leq \beta$ **then**

 | Badany wierzchołek jest oznaczany jako “przetworzony”;

end

if $P(\Theta) > \beta$ **then**

 | Dla każdego atrybutu $f_i \in E(\Theta)$, rysowana jest wiązka krawędzi wychodzących z wierzchołka Θ . Niech $E^*(\Theta, f_i) = \{b_1, \dots, b_t\}$ (należy zauważyć, że $t \geq 2$ tylko dla minimalnego atrybutu z Θ), wówczas rysowanych jest t krawędzi wychodzących z wierzchołka Θ . Krawędzie te są oznaczane odpowiednio parami $(f_i, b_1), \dots, (f_i, b_t)$. Krawędzie te wchodzą do wierzchołków $\Theta(f_i, b_1), \dots, \Theta(f_i, b_t)$. Jeśli jakieś wierzchołki $\Theta(f_i, b_1), \dots, \Theta(f_i, b_t)$ nie istnieją w grafie, wówczas są dodawane do grafu. Każdemu wierszowi r tabeli Θ zostaje przypisany zbiór atrybutów $E_{\Delta_\beta^*(T)}(\Theta, r) \subseteq E(\Theta)$. Wierzchołek Θ jest oznaczany jako “przetworzony”;

end

end

return Graf $\Delta_\beta^*(T)$;

które nie są stałe w T (posiadają przynajmniej dwie różne wartości). Dla każdego atrybutu $f_i \in E(T)$, definiowany jest zbiór wartości $E^*(T, f_i)$. Jeśli f_i jest atrybutem o minimalnej liczbie wartości i posiada minimalny indeks i spośród takich atrybutów (atrybut ten będzie nazywany *minimalnym atrybutem dla T*), wówczas $E^*(T, f_i)$ jest zbiorem wszystkich wartości f_i w T . W przeciwnym przypadku, $E^*(T, f_i)$ zawiera tylko najczęstszą wartość f_i w T .

Algorytm 5 przedstawia algorytm, który konstruuje graf $\Delta_\beta^*(T)$. Jest to skierowany graf acykliczny. Wierzchołkami tego grafu są separowalne podtablele tabeli T . Wierzchołek grafu jest nazywany *końcowy*, jeśli nie wychodzą z niego żadne krawędzie. Należy zauważyć, że wierzchołek Θ jest końcowy w grafie $\Delta_\beta^*(T)$ wtedy i tylko wtedy, jeśli $P(\Theta) \leq \beta$.

W dalszej części rozdziału, zostanie opisana procedura optymalizacji grafu $\Delta_\beta^*(T)$ względem pokrycia. Wynikiem pracy tej procedury jest graf G , który zawiera ten sam zbiór wierzchołków i krawędzi jak graf $\Delta_\beta^*(T)$. Różnicą jest to, że dowolny wiersz r w każdym niekończącym wierzchołku Θ w grafie G ma przypisany zbiór atrybutów $E_G(\Theta, r) \subseteq E_{\Delta_\beta^*(T)}(\Theta, r)$. Jest także możliwe, że $G = \Delta_\beta^*(T)$.

Poniżej, dla każdego wierzchołka Θ w grafie G i dla każdego wiersza r z Θ , zostanie opisany zbiór reguł decyzyjnych $Rul_G(\Theta, r)$. W tym celu, należy prze-

chodzić w grafie G zaczynając od końcowych wierzchołków grafu, do wierzchołka T .

Niech Θ będzie końcowym wierzchołkiem w grafie G i d będzie najbardziej wspólną decyzją dla Θ . Wówczas $Rul_G(\Theta, r) = \{\rightarrow d\}$.

Niech teraz Θ będzie niekończącym wierzchołkiem w grafie G takim, że dla każdego potomka Θ' wierzchołka Θ i dla każdego wiersza r' tabeli Θ' , zbiór reguł $Rul_G(\Theta', r')$ został zdefiniowany. Niech $r = (b_1, \dots, b_n)$ będzie wierszem tabeli Θ . Dla dowolnego atrybutu $f_i \in E_G(\Theta, r)$, jest definiowany zbiór reguł $Rul_G(\Theta, r, f_i)$: $Rul_G(\Theta, r, f_i) = \{f_i = b_i \wedge \gamma \rightarrow s : \gamma \rightarrow s \in Rul_G(\Theta(f_i, b_i), r)\}$. Wówczas $Rul_G(\Theta, r) = \bigcup_{f_i \in E_G(\Theta, r)} Rul_G(\Theta, r, f_i)$.

Procedura optymalizacji względem pokrycia c

Niech $G = \Delta_\beta^*(T)$. W wyniku pracy procedury optymalizacji grafu G względem pokrycia, każdy wiersz r każdego wierzchołka Θ w grafie G , będzie miał przypisaną liczbę $Opt_G^c(\Theta, r)$ – maksymalne pokrycie reguły decyzyjnej ze zbioru $Rul_G(\Theta, r)$, i dla każdego wiersza r w każdym niekończącym wierzchołku Θ grafu G , zostanie zmieniony zbiór $E_G(\Theta, r)$ przypisany do r w Θ . Uzyskany graf jest oznaczany przez G^c .

Niech Θ będzie końcowym wierzchołkiem w grafie G i d będzie najbardziej wspólną decyzją dla Θ . Wówczas, do każdego wiersza r tabeli Θ jest przypisywana liczba $Opt_G^c(\Theta, r)$ która jest równa liczbie wierszy w Θ , które mają przypisaną decyzję d .

Niech Θ będzie niekończącym wierzchołkiem w grafie G i wszyscy potomkowie Θ zostali przetworzeni. Niech $r = (b_1, \dots, b_n)$ będzie wierszem tabeli Θ . Do wiersza r w tabeli Θ jest przypisywana liczba $Opt_G^c(\Theta, r) = \max\{Opt_G^c(\Theta(f_i, b_i), r) : f_i \in E_G(\Theta, r)\}$ i zbiór $E_{G^c}(\Theta, r) = \{f_i : f_i \in E_G(\Theta, r), Opt_G^c(\Theta(f_i, b_i), r) = Opt_G^c(\Theta, r)\}$.

Można pokazać, że $Rul_{G^c}(\Theta, r)$ jest zbiorem wszystkich reguł z $Rul_G(\Theta, r)$ o maksymalnym pokryciu.

Wyniki eksperymentów

Tabela 3 przedstawia średnie wartości maksymalnego pokrycia β -reguł decyzyjnych dla tablic decyzyjnych umieszczonych w Repozytorium Uczenia Maszynowego [11]. Niech T będzie jedną z tych tablic decyzyjnych. Dla tabeli T zostały zastosowane wartości β ze zbioru $B(T) = \{P(T) \times 0.15, P(T) \times 0.25, P(T) \times 0.35\}$. Niech $\beta \in B(T)$. Dla każdego wiersza r w T , została uzyskana maksymalna wartość pokrycia β -reguły decyzyjnej dla T i r . Następnie, dla wierszy w T została obliczona średnia wartość pokrycia reguł o maksymalnym pokryciu - jedna reguła dla każdego wiersza. Wyniki zostały umieszczone w kolumnie *avg*. W celu porównania średnich wartości pokrycia β -reguł decyzyjnych konstruowanych przez algorytm dynamicznego programowania przedstawiony w poprzednim rozdziale zostały wykonane eksperymenty, których wyniki zawarte są w kolumnie *avg-dp*. Ostatnia kolumna *rel diff* przedstawia relatywną różnicę, która jest równa $(Opt_Coverage - Coverage)/Opt_Coverage$, gdzie *Coverage* oznacza

średnie pokrycie β -reguł decyzyjnych konstruowanych przez proponowany algorytm, $Opt_Coverage$ oznacza średnie pokrycie β -reguł decyzyjnych konstruowanych przez algorytm dynamicznego programowania. Ostatni wiersz w Tabeli 3 przedstawia średnią wartość relatywnej różnicy dla rozważanych tablic decyzyjnych.

Przedstawione wyniki pokazują, że średnia relatywna różnica maleje wraz ze wzrostem wartości β . Dla $\beta = P(T) \times 0.35$, relatywna różnica wynosi 0 dla dziesięciu tablic decyzyjnych. Dla tablic decyzyjnych “lymphography”, “soybean-small”, “teeth” i “zoo-data”, dla każdego $\beta \in B(T)$, relatywna różnica wynosi 0.

Tabela 3: Średnie pokrycie β -reguł decyzyjnych

Tablica decyzyjna	Wiersze	Attr	$\beta = P(T) \times 0.15$			$\beta = P(T) \times 0.25$			$\beta = P(T) \times 0.35$		
			avg	avg-dp	rel diff	avg	avg-dp	rel diff	avg	avg-dp	rel diff
adult-stretch	16	4	6.75	7	0.04	7.5	7.5	0	7.5	7.5	0
balance-scale	625	4	88.48	92.31	0.04	88.48	92.31	0.04	88.48	92.31	0.04
breast-cancer	266	9	81.2	81.84	0.01	114.9	117.39	0.02	124.35	125.08	0.01
cars	1728	6	486.54	499.26	0.03	486.54	499.26	0.03	486.54	499.26	0.03
house-votes	279	16	148.76	150.33	0.01	156.69	157.54	0.01	160.15	160.55	0
lymphography	148	18	56.33	56.35	0	57.94	57.94	0	59.06	59.06	0
monks-1-test	432	6	77	81	0.05	108	108	0	108	108	0
monks-1-train	124	6	25.12	28.01	0.1	28.36	30.22	0.06	34.4	34.4	0
monks-2-test	432	6	95.3	100.74	0.05	135.11	136.19	0.01	146.5	146.5	0
monks-2-train	169	6	34.43	37.39	0.08	41.91	49.83	0.16	52.54	54.23	0.03
monks-3-test	432	6	97.11	116	0.16	129	120	0	120	120	0
monks-3-train	122	6	24.07	32.15	0.25	27.75	33.52	0.17	33	35.04	0.06
nursery	12960	8	2511.42	3066.1	0.18	2705.29	3066.1	0.12	2917.33	3066.1	0.05
soybean-small	47	35	15.64	15.64	0	17	17	0	17	17	0
teeth	23	8	1	1	0	1	1	0	1	1	0
zoo-data	59	16	16.85	16.86	0	17.83	17.83	0	18.85	18.85	0
average					0.06			0.04			0.01

Badany był także rozmiar skierowanego grafu acyklicznego tj. liczba wierzchołków i liczba krawędzi w grafie konstruowanym przez proponowany algorytm. Wyniki eksperymentów zawarte w pracy [65] pokazują, że rozmiar skierowanego grafu acyklicznego konstruowanego przez proponowany algorytm jest mniejszy niż rozmiar skierowanego grafu acyklicznego konstruowanego przez algorytm dynamicznego programowania. W szczególności, dla tablic decyzyjnych “lymphography”, “soybean-small”, “teeth” i “zoo-data”, średnie wartości pokrycia reguł są takie same (zobacz Tabela 3) ale istnieje różnica odnośnie liczby wierzchołków (więcej niż jeden raz) oraz odnośnie liczby krawędzi (więcej niż dwa razy).

6.3 Powiązania pomiędzy długością i pokryciem reguł decyzyjnych

Rozdział ten, na podstawie pracy [10], poświęcony jest badaniu powiązań pomiędzy długością i pokryciem nieredukowalnych 0-reguł decyzyjnych dla danego wiersza tablicy decyzyjnej T oraz powiązań pomiędzy długością i pokryciem pełnych systemów reguł decyzyjnych dla tablicy decyzyjnej T . System S prawdziwych reguł dla T jest nazywany pełny dla T , jeśli istnieje odpowiedniość jeden-do-jeden pomiędzy regułami z S i wierszami z T taka, że dla każdego wiersza r , odpowiadająca mu reguła jest zrealizowana dla r .

Powiązania pomiędzy długością i pokryciem reguł decyzyjnych dla danego wiersza

W tej części rozdziału zostanie opisane narzędzie dla badania powiązań pomiędzy długością i pokryciem dokładnych reguł decyzyjnych, dla danego wiersza. Narzędzie to wykorzystuje podejście oparte na dynamicznym programowaniu.

Niech T będzie tablicą decyzyjną z n kolumnami oznaczonymi przez atrybuty f_1, \dots, f_n i $r = (b_1, \dots, b_n)$ będzie wierszem z T . Niech Θ będzie wierzchołkiem grafu $\Delta = \Delta_0(T)$ zawierającym wiersz r .

Z Twierdzenia 32 wynika, że zbiór $Rul_{\Delta}(\Theta, r)$ jest równy zbiorowi wszystkich nieredukowalnych reguł decyzyjnych (0-reguł decyzyjnych) dla Θ i r .

Poniżej zostanie opisane powiązanie pomiędzy pokryciem i długością nieredukowalnych reguł decyzyjnych dla danego wiersza r tablicy decyzyjnej T .

Przez $l_{min}(\Theta, r)$ jest oznaczana minimalna długość reguły ze zbioru $Rul_{\Delta}(\Theta, r)$. Niech $B_{\Theta, r} = \{l_{min}(\Theta, r), l_{min}(\Theta, r) + 1, \dots, n\}$. Funkcja $F_{\Theta, r}$ jest definiowana następująco, $F_{\Theta, r} : B_{\Theta, r} \rightarrow \mathbb{N}$, gdzie \mathbb{N} jest zbiorem liczb naturalnych. Dla dowolnego $m \in B_{\Theta, r}$, $F_{\Theta, r}(m) = \max\{c(\tau) : \tau \in Rul_{\Delta}(\Theta, r), l(\tau) \leq m\}$. Funkcja ta opisuje powiązanie pomiędzy pokryciem i długością reguł decyzyjnych: $F_{\Theta, r}(m)$ jest równa maksymalnej wartości pokrycia spośród wszystkich nieredukowalnych reguł decyzyjnych dla Θ i r , których długość jest najwyżej m . Funkcja $F_{\Theta, r}$ może być reprezentowana przez n-tkę $(F_{\Theta, r}(t), F_{\Theta, r}(t+1), \dots, F_{\Theta, r}(n))$, gdzie $t = l_{min}(\Theta, r)$.

Teraz zostanie opisana procedura, która pozwala znaleźć (opisać) funkcję $F_{\Theta, r}$, dla każdego wierzchołka Θ w grafie Δ , który zawiera wiersz r .

Niech Θ będzie zdegenerowaną tabelą, która zawiera wiersz r . Wszystkie wiersze Θ są oznaczone przez tą samą decyzję d . Wiadomo, że istnieje dokładnie jedna nieredukowalna reguła decyzyjna $\rightarrow d$ dla Θ i r . Zatem, $l_{min}(\Theta, r) = 0$ i $F_{\Theta, r}(m) = N(\Theta)$ dla dowolnego $m \in B_{\Theta, r}$.

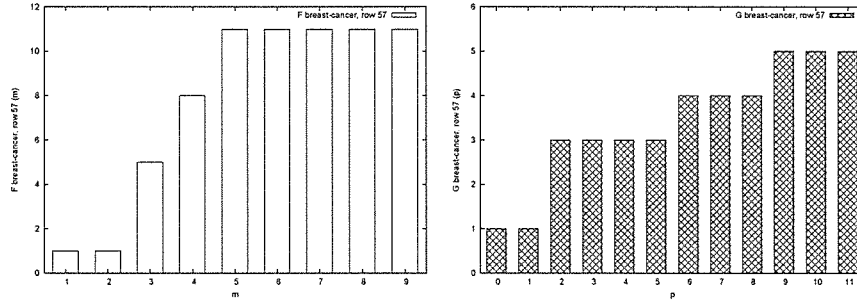
Niech Θ będzie niezdegenerowaną tabelą (wierzchołkiem) w grafie Δ zawierającą wiersz r . Wiadomo, że $Rul_{\Delta}(\Theta, r) = \bigcup_{f_i \in E(\Theta)} Rul_{\Delta}(\Theta, r, f_i)$, gdzie $Rul_{\Delta}(\Theta, r, f_i) = \{f_i = b_i \wedge \alpha \rightarrow d : \alpha \rightarrow d \in Rul_{\Delta}(\Theta(f_i, b_i), r)\}$. Stąd wynika, że $l_{min}(\Theta, r) = \min\{l_{min}(\Theta(f_i, b_i), r) : f_i \in E(\Theta)\} + 1$ i dla dowolnego $m \in B_{\Theta, r}$, $F_{\Theta, r}(m) = \max\{F_{\Theta(f_i, b_i), r}(m-1) : f_i \in E(\Theta), m-1 \geq l_{min}(\Theta(f_i, b_i), r)\}$. Został tutaj wykorzystany fakt, że długość reguły $f_i = b_i \wedge \alpha \rightarrow d$ jest równa długości reguły $\alpha \rightarrow d$ plus 1, a pokrycie reguły $f_i = b_i \wedge \alpha \rightarrow d$ dla tabeli Θ jest równe pokryciu reguły $\alpha \rightarrow d$ dla tabeli $\Theta(f_i, b_i)$.

Teraz zostanie opisane odwrotne powiązanie, tj. powiązanie pomiędzy długością i pokryciem nieredukowalnych reguł decyzyjnych, dla danego wiersza r tablicy decyzyjnej T . Niech T będzie tablicą decyzyjną zawierającą n kolumn oznaczonych przez atrybuty f_1, \dots, f_n i $r = (b_1, \dots, b_n)$ będzie wierszem w T . Przez $c_{max}(T, r)$ jest oznaczane maksymalne pokrycie reguły decyzyjnej z $Rul_{\Delta}(T, r)$. Niech $C_{T, r} = \{0, 1, \dots, c_{max}(T, r)\}$. Funkcja $G_{T, r}$ definiowana jest następująco $G_{T, r} : C_{T, r} \rightarrow \mathbb{Z}^+$, gdzie \mathbb{Z}^+ jest zbiorem nieujemnych liczb całkowitych. Dla dowolnego $p \in C_{T, r}$, $G_{T, r}(p) = \min\{l(\tau) : \tau \in Rul(T, r), c(\tau) \geq p\}$. Wartość $G_{T, r}(p)$ jest równa minimalnej długości spośród wszystkich nieredukowalnych reguł decyzyjnych dla T i r , dla których wartość pokrycia wynosi co naj-

mniej p . W celu znalezienia tej wartości została użyta funkcja $F_{T,r}$. Można pokazać, że $G_{T,r}(p) = \min\{m : m \in \{l_{\min}(T,r), \dots, n\}, F_{T,r}(m) \geq p\}$. Oznaczamy $m_0 = \min\{m : m \in \{l_{\min}(T,r), \dots, n\}, F_{T,r}(m) \geq p\}$. Wiadomo, że istnieje reguła $\tau \in Rul_{\Delta}(T,r)$ taka, że $l(\tau) \leq m_0$ i $c(\tau) \geq p$, oraz nie istnieje reguła $\rho \in Rul_{\Delta}(T,r)$ taka, że $l(\rho) < m_0$ i $c(\rho) \geq p$. Zatem wynika, że $G_{T,r}(p) = m_0$.

Należy dodać, że $c_{\max}(T,r) = F_{T,r}(n)$. Wiadomo, że $F_{T,r}$ jest funkcją niemalejącą. Zatem, w celu znalezienia wartości m_0 zostało wykorzystane wyszukiwanie binarne, które wymaga $O(\log n)$ porównań.

W celu zilustrowania opisanych powiązań pomiędzy długością i pokryciem nieredukowalnych reguł decyzyjnych zbadane zostały tablice decyzyjne umieszczone w Repozytorium Uczenia Maszynowego [11]. Funkcje przedstawione na



Rysunek 5: Funkcje $F_{T,r}(m)$ i $G_{T,r}(p)$ dla wiersza 57 tablicy decyzyjnej “breast-cancer”

Rysunku 5, $F_{\text{breast-cancerrow}57}$ i $G_{\text{breast-cancerrow}57}$ są rosnące, co oznacza, że dla wiersza 57 tablicy decyzyjnej “breast-cancer” nie istnieją całkowicie optymalne reguły względem długości i pokrycia.

Powiązania pomiędzy długością i pokryciem dla systemów reguł decyzyjnych

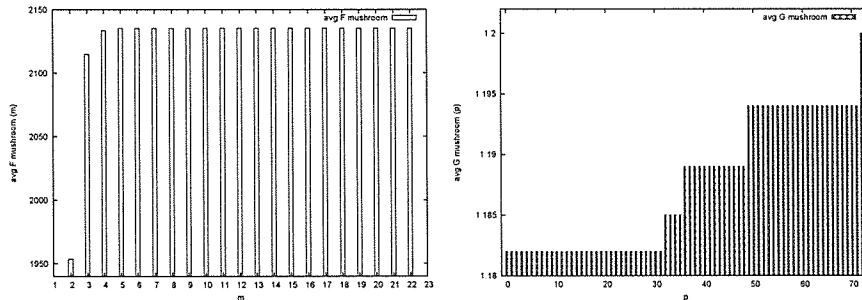
Poniżej zostaną opisane powiązania pomiędzy długością i pokryciem dla pełnych systemów reguł decyzyjnych dla tablicy decyzyjnej T . Przez $Row(T)$ jest oznaczony zbiór wierszy z T , $|Row(T)| = N(T)$. Niech S będzie systemem zawierającym $N(T)$ reguł decyzyjnych dla T gdzie, dla każdego wiersza $r \in Row(T)$, istnieje reguła decyzyjna $rule$, która jest nieredukowalną regułą decyzyjną dla T i r . System S może zawierać takie same reguły. Przez $\mathcal{P}(T)$ jest oznaczany zbiór wszystkich takich systemów, które są nazywane *pełnymi dla T systemami reguł decyzyjnych*.

Teraz zostanie opisane powiązanie pomiędzy pokryciem i długością reguł dla systemów $S \in \mathcal{P}(T)$. Przez $l_{\max}(S)$ jest oznaczana maksymalna długość reguły z S , przez $c_{\text{avg}}(S)$ – średnie pokrycie reguł z S : $c_{\text{avg}}(S) = \frac{\sum_{rule \in S} c(rule)}{|Row(T)|}$.

Niech $B_T = \{l_{min}(T), \dots, n\}$, gdzie n jest liczbą atrybutów warunkowych w T i $l_{min}(T) = \max\{l_{min}(T, r) : r \in Row(T)\}$. Celem jest znalezienie, dla każdego $m \in B_T$, wartości $F_T^{avg}(m) = \max\{c_{avg}(S) : S \in \mathcal{P}(T), l_{max}(S) \leq m\}$, która jest równa maksymalnej wartości średniego pokrycia reguł spośród wszystkich pełnych systemów $S \in \mathcal{P}(T)$, gdzie długość każdej reguły wynosi najwyżej m . Wartość $F_T^{avg}(m)$ jest obliczana w następujący sposób: $F_T^{avg}(m) = \frac{\sum_{r \in Row(T)} F_{T,r}(m)}{|Row(T)|}$.

Teraz zostanie opisane odwrotne powiązanie, tj. pomiędzy długością i pokryciem dla systemów $S \in \mathcal{P}(T)$. Przez $c_{min}(S)$ jest oznaczana minimalna wartość pokrycia reguły z S . Przez $l_{avg}(S)$ jest oznaczana średnia długość reguł z S : $l_{avg}(S) = \frac{\sum_{rule \in S} l(rule)}{|Row(T)|}$. Niech $C_T = \{0, 1, \dots, c_{max}(T)\}$, gdzie $c_{max}(T) = \min\{c_{max}(T, r) : r \in Row(T)\}$. Celem jest znalezienie, dla każdego $p \in C_T$, wartości $G_T^{avg}(p) = \min\{l_{avg}(S) : S \in \mathcal{P}(T), c_{min}(S) \geq p\}$, która jest równa minimalnej wartości średniej długości reguł spośród pełnych systemów $S \in \mathcal{P}(T)$, gdzie pokrycie każdej reguły wynosi co najmniej p . Wartość $G_T^{avg}(p)$ jest obliczana w następujący sposób: $G_T^{avg}(p) = \frac{\sum_{r \in Row(T)} G_{T,r}(p)}{|Row(T)|}$.

Rysunek 6 przedstawia opisane powiązania pomiędzy długością i pokryciem dla pełnych systemów reguł decyzyjnych, dla tablicy decyzyjnej "mushroom" z Repozytorium Uczenia Maszynowego [11].



Rysunek 6: Funkcje $F_T^{avg}(m)$ i $G_T^{avg}(p)$ dla tablicy decyzyjnej "mushroom"

Przedstawione narzędzie do badania powiązań pomiędzy długością i pokryciem reguł decyzyjnych może zostać wykorzystane podczas tworzenia klasyfikatorów. Poza tym, krótkie reguły, które pokrywają wiele obiektów, są ważne z punktu widzenia reprezentacji wiedzy.

7 Podejście zachłanne i podejście oparte na dynamicznym programowaniu dla optymalizacji reguł asocjacyjnych

Rozdział ten, w oparciu o prace [4, 5, 38, 39, 67], poświęcony jest badaniu częściowych (przybliżonych) reguł asocjacyjnych. Częściowe reguły asocjacyjne mogą być stosowane jako sposób reprezentacji wiedzy, w procesie wnioskowania w systemach ekspertowych, do konstrukcji klasyfikatorów oraz do wypełniania brakujących wartości atrybutów. Rozdział ten składa się z pięciu części.

W pierwszej części (rozdział 7.1) badane są aproksymacyjne algorytmy dla minimalizacji długości częściowych reguł asocjacyjnych. Druga część (rozdział 7.2) poświęcona jest badaniu algorytmów aproksymacyjnych dla minimalizacji całkowitej wagi atrybutów występujących w częściowych regułach asocjacyjnych. Badane są głównie algorytmy zachłanne z wagami dla konstruowania reguł. Ten rozdział zawiera także granice dokładności rozważanych algorytmów oraz granice minimalnej wagi częściowych reguł asocjacyjnych, na podstawie danych uzyskanych podczas pracy algorytmu zachłannego. W rozdziałach 7.1 i 7.2, w dowodach niektórych twierdzeń, zostały użyte wyniki z prac [24, 34, 36, 44, 53–56]. Trzecia część (rozdział 7.3) zawiera wprowadzenie do algorytmów dynamicznego programowania dla optymalizacji reguł asocjacyjnych. Przedstawione wyniki związane są z maksymalizacją pokrycia reguł. Czwarta część (rozdział 7.4) poświęcona jest zastosowaniu podejścia opartego na dynamicznym programowaniu dla konstruowania systemów reguł asocjacyjnych o stosunkowo dobrej jakości, które pokrywają większą część wierszy. Liczebność konstruowanych systemów reguł jest rozsądna. W piątej części (rozdział 7.5) zostało porównanych pięć heurystyk zachłannych dla konstruowania reguł asocjacyjnych. Stosując algorytmy dynamicznego programowania porównano konstruowane reguły z regułami o minimalnej długości lub maksymalnym pokryciu. Wynikiem jest wybór dwóch heurystyk dla minimalizacji długości i maksymalizacji pokrycia reguł.

7.1 Algorytm zachłanny dla konstruowania częściowych reguł asocjacyjnych

W tej części rozdziału, na podstawie pracy [38], badane są algorytmy wielomianowe (głównie algorytm zachłanny) dla konstruowania częściowych (przybliżonych) reguł asocjacyjnych oraz granice długości konstruowanych reguł. Podejście to różni się od znanego podejścia opartego na zbiorach częstych [2, 18, 58]. Badany jest specjalny rodzaj reguł asocjacyjnych. Zamiast standardowych parametrów reguł asocjacyjnych, takich jak wsparcie i zaufanie [2], badana jest długość reguł asocjacyjnych.

Podstawowe pojęcia

System informacyjny I jest tabelą zawierającą n wierszy (odpowiadających obiektom) i m kolumn oznaczonych przez atrybuty f_1, \dots, f_m . Tabela ta wypełniona jest przez nieujemne liczby całkowite (wartości atrybutów). Jeśli tabela wypełniona jest przez liczby ze zbioru $\{0, 1\}$, wówczas system informacyjny I

nazywany jest *binary*. Formalnie, system informacyjny jest definiowany jako para $I = (U, A)$, gdzie U jest niepustym, skończonym zbiorem obiektów (wierszy) określanym jako *uniwersum*, A jest niepustym, skończonym zbiorem atrybutów. Każdy $f \in A$ jest funkcją $f : U \rightarrow V_f$, V_f jest zbiorem wartości atrybutu f .

Niech $r = (b_1, \dots, b_m)$ będzie wierszem w I i f_p będzie atrybutem ze zbioru $\{f_1, \dots, f_m\}$. Przez $U(I, r, f_p)$ jest oznaczany zbiór wierszy z I , które są różne od wiersza r w kolumnie f_p i przynajmniej jednej kolumnie f_j takiej, że $j \in \{1, \dots, m\} \setminus \{p\}$. Atrybut f_i *separuje* wiersz $r' \in U(I, r, f_p)$ od wiersza r , jeśli wiersze r i r' posiadają różne wartości na przecięciu z kolumną f_i . Przez $U(I, r, f_p, f_i)$ jest oznaczany zbiór wierszy $r' \in U(I, r, f_p)$, które atrybut f_i separuje od wiersza r . Trójka (I, r, f_p) jest nazywana *problemem reguły asocjacyjnej*. Niech $\alpha \in \mathbb{R}$ i $0 \leq \alpha < 1$. Reguła

$$(f_{i_1} = b_{i_1}) \wedge \dots \wedge (f_{i_t} = b_{i_t}) \rightarrow f_p = b_p \quad (8)$$

jest nazywana α -regułą asocjacyjną dla (I, r, f_p) jeśli $i_1, \dots, i_t \in \{1, \dots, m\} \setminus \{p\}$ oraz atrybuty f_{i_1}, \dots, f_{i_t} separują od r przynajmniej $(1 - \alpha)|U(I, r, f_p)|$ wierszy ze zbioru $U(I, r, f_p)$ (takie reguły są także nazywane *częściowymi* regułami asocjacyjnymi). Liczba t oznacza *długość* rozważanej reguły asocjacyjnej. Jeśli $U(I, r, f_p) = \emptyset$, wówczas dla dowolnego $i_1, \dots, i_t \in \{1, \dots, m\} \setminus \{p\}$, reguła (8) jest α -regułą asocjacyjną dla (I, r, f_p) . Reguła (8) z pustą lewą stroną (kiedy $t = 0$) jest α -regułą asocjacyjną dla (I, r, f_p) jeśli $U(I, r, f_p) = \emptyset$. Na przykład, 0.01-reguła asocjacyjna oznacza, że należy odseparować od wiersza r przynajmniej 99% wierszy z $U(I, r, f_p)$. Przez $L_{\min}(\alpha) = L_{\min}(\alpha, I, r, f_p)$ jest oznaczana minimalna długość α -reguły asocjacyjnej dla (I, r, f_p) .

Dokładność algorytmu zachłannego

Algorytm 6 przedstawia *algorytm zachłanny z progiem* α , który konstruuje α -regułę asocjacyjną dla (I, r, f_p) .

Algorytm 6: Algorytm zachłanny dla konstruowania częściowej reguły asocjacyjnej

Wejście: System informacyjny I z atrybutami f_1, \dots, f_m , wiersz $r = (b_1, \dots, b_m)$ w I , atrybut $f_p \in \{f_1, \dots, f_m\}$, liczba rzeczywista α , $0 \leq \alpha < 1$.

Wyjście: α -reguła asocjacyjna dla (I, r, f_p) .

$Q \leftarrow \emptyset$;

while atrybuty z Q separują od r mniej niż $(1 - \alpha)|U(I, r, f_p)|$ wierszy z

$U(I, r, f_p)$ **do**

wybierz $f_i \in \{f_1, \dots, f_m\} \setminus \{f_p\}$ o minimalnym indeksie i taki, że f_i separuje od r maksymalną liczbę wierszy z $U(I, r, f_p)$ nieodseparowanych przez atrybuty z Q ;

$Q \leftarrow Q \cup \{f_i\}$;

end

return $\bigwedge_{f_i \in Q} (f_i = b_i) \rightarrow f_p = b_p$;

Przez $L_{\text{greedy}}(\alpha) = L_{\text{greedy}}(\alpha, I, r, f_p)$ jest oznaczana długość konstruowanej α -reguły asocjacyjnej dla (I, r, f_p) . Niech I będzie systemem informacyjnym zawierającym m kolumn oznaczonych przez atrybuty f_1, \dots, f_m , r będzie wierszem w I i $f_p \in \{f_1, \dots, f_m\}$.

Twierdzenie 36. *Niech $0 \leq \alpha < 1$ i $\lceil (1 - \alpha)|U(I, r, f_p)| \rceil \geq 2$. Wówczas*

$$L_{\text{greedy}}(\alpha) < L_{\min}(\alpha)(\ln \lceil (1 - \alpha)|U(I, r, f_p)| \rceil - \ln \ln \lceil (1 - \alpha)|U(I, r, f_p)| \rceil + 0.78).$$

Twierdzenie 37. *Niech $0 \leq \alpha < 1$. Wówczas, dla dowolnej liczby naturalnej $t \geq 2$, istnieje problem reguły asocjacyjnej (I, r, f_p) taki, że $\lceil (1 - \alpha)|U(I, r, f_p)| \rceil = t$ i $L_{\text{greedy}}(\alpha) > L_{\min}(\alpha)(\ln \lceil (1 - \alpha)|U(I, r, f_p)| \rceil - \ln \ln \lceil (1 - \alpha)|U(I, r, f_p)| \rceil - 0.31)$.*

Twierdzenie 38. *Niech $0 \leq \alpha < 1$ i $U(I, r, f_p) \neq \emptyset$. Wówczas*

$$L_{\text{greedy}}(\alpha) \leq L_{\min}(\alpha) \left(1 + \ln \left(\max_{j \in \{1, \dots, m\} \setminus \{p\}} |U(I, r, f_p, f_j)| \right) \right).$$

Wielomianowe algorytmy aproksymacyjne

Poniższe twierdzenia dotyczące wielomianowych algorytmów aproksymacyjnych dla optymalizacji reguł zostały udowodnione, na podstawie wyników badań dotyczących problemu pokrycia zbioru.

Twierdzenie 39. *Niech $0 \leq \alpha < 1$. Wówczas problem konstruowania α -reguły asocjacyjnej o minimalnej długości jest NP-trudny.*

Twierdzenie 40. *Niech $\alpha \in \mathbb{R}$ i $0 \leq \alpha < 1$. Jeśli $NP \not\subseteq DTIME(n^{O(\log \log n)})$ wówczas, dla dowolnego ε , $0 < \varepsilon < 1$, nie istnieje algorytm o złożoności wielomianowej taki, że dla danego problemu reguły asocjacyjnej (I, r, f_p) z $U(I, r, f_p) \neq \emptyset$, konstruuje α -regulę asocjacyjną dla (I, r, f_p) , której długość wynosi najwyżej $(1 - \varepsilon)L_{\min}(\alpha, I, r, f_p) \ln |U(I, r, f_p)|$.*

Granice $L_{\min}(\alpha)$ uzyskane na podstawie danych o pracy algorytmu zachłannego

Na podstawie danych o pracy algorytmu zachłannego została znaleziona najlepsza dolna granica $L_{\text{greedy}}(\alpha) = L_{\text{greedy}}(\alpha, I, r, f_p)$ zależna od tych danych. Niech algorytm zachłanny z progiem α zostanie zastosowany do problemu (I, r, f_p) . Niech podczas konstruowania α -reguły asocjacyjnej algorytm zachłanny wybiera atrybuty f_{j_1}, \dots, f_{j_t} . $U(I, r, f_p, f_{j_0}) = \emptyset$, $\delta_0 = 0$ i dla $i = 1, \dots, t$, $\delta_i = |U(I, r, f_p, f_{j_i}) \setminus (U(I, r, f_p, f_{j_0}) \cup \dots \cup U(I, r, f_p, f_{j_{i-1}}))|$. Jako dane o pracy algorytmu zachłannego została użyta n -tka $(\delta_1, \dots, \delta_t)$ oraz liczby $n = |U(I, r, f_p)|$ i α . Przyjmijmy oznaczenie

$$l(\alpha, n, \delta_1, \dots, \delta_t) = \max \left\{ \left\lceil \frac{\lceil (1 - \alpha)n \rceil - (\delta_0 + \dots + \delta_i)}{\delta_{i+1}} \right\rceil : i = 0, \dots, t - 1 \right\}.$$

Twierdzenie 41. *Parameter $l(\alpha, n, \delta_1, \dots, \delta_t)$ jest najlepszą dolną granicą $L_{\min}(\alpha)$ zależną od $(\delta_1, \dots, \delta_t)$, n , i α .*

Niech $l(\alpha) = l(\alpha, n, \delta_1, \dots, \delta_t)$. Wartość $l(\alpha)$ może zostać użyta do uzyskania górnych granic długości częściowych reguł asocjacyjnych konstruowanych przez algorytm zachłanny. Oznacza to, że $l(\alpha)$ jest nietrywialną dolną granicą minimalnej długości częściowych reguł asocjacyjnych.

Twierdzenie 42. *Niech α i β będą liczbami rzeczywistymi takimi, że $0 < \beta \leq \alpha < 1$. Wówczas $L_{\text{greedy}}(\alpha) < l(\alpha - \beta) \ln \left(\frac{1 - \alpha + \beta}{\beta} \right) + 1$.*

Górna granica $L_{\text{greedy}}(\alpha)$

Została uzyskana górna granica $L_{\text{greedy}}(\alpha) = L_{\text{greedy}}(\alpha, I, r, f_p)$, która nie zależy od $|U(I, r, f_p)|$ i jest w pewnym sensie nieulepszalna.

Twierdzenie 43. *Niech α i β będą liczbami rzeczywistymi takimi, że $0 < \beta \leq \alpha < 1$. Wówczas $L_{\text{greedy}}(\alpha) < L_{\min}(\alpha - \beta) \ln \left(\frac{1 - \alpha + \beta}{\beta} \right) + 1$.*

Twierdzenie 44. *Nie istnieje liczba rzeczywista $\delta < 1$ taka, że dla dowolnego problemu reguły asocjacyjnej (I, r, f_p) i dla dowolnego rzeczywistego α i β , $0 < \beta \leq \alpha < 1$, zachodzi: $L_{\text{greedy}}(\alpha) \leq \delta \left(L_{\min}(\alpha - \beta) \ln \left(\frac{1 - \alpha + \beta}{\beta} \right) + 1 \right)$.*

Reguły asocjacyjne dla większości binarnych systemów informacyjnych

Zostało udowodnione istnienie krótkich reguł dla większości binarnych systemów informacyjnych.

Twierdzenie 45. *Niech będą badane binarne systemy informacyjne zawierające n wierszy i m kolumn oznaczonych przez atrybuty f_1, \dots, f_m . Niech t będzie liczbą naturalną, $n \geq t + 4$, $m \geq (t + 4) \lceil \log_2 n \rceil$ i $k = (t + 4) \lceil \log_2 n \rceil$. Wówczas część systemów informacyjnych I , w których dla dowolnego wiersza r i dowolnego atrybutu f_p istnieje 0-reguła asocjacyjna dla (I, r, f_p) o długości najwyżej k , wynosi $1 - \frac{1}{n^t}$.*

W wyniku badań uzyskano pewne potwierdzenia nieformalnej tzw. 0.5-hipotezy dla reguł asocjacyjnych: dla większości systemów informacyjnych I dla każdego wiersza r i każdego atrybutu f_p , podczas konstruowania częściowej reguły asocjacyjnej dla (I, r, f_p) , w każdym kroku algorytm zachłanny wybiera atrybut, który separuje od wiersza r przynajmniej połowę nieodseparowanych wcześniej wierszy ze zbioru $U(I, r, f_p)$. W tym przypadku, $L_{\text{greedy}}(\alpha) \leq \lceil \log_2 \left(\frac{1}{\alpha} \right) \rceil$, dla dowolnego rzeczywistego α , $0 < \alpha < 1$.

Twierdzenie 46. *Niech będą badane binarne systemy informacyjne z n wierszami i m kolumnami oznaczonymi przez atrybuty f_1, \dots, f_m . Niech $m \geq n + \log_2 n + \log_2 m$. Wówczas część systemów informacyjnych, dla których zachodzi 0.5-hipoteza dla reguł asocjacyjnych, wynosi co najmniej $1 - \frac{1}{2^t}$, gdzie $t = m - (n + \log_2 n + \log_2 m)$.*

Uzyskane wyniki eksperymentalne potwierdziły 0.5-hipotezę, nie tylko dla przypadku, kiedy $m > n$.

7.2 Algorytmy zachłanne z wagami dla konstruowania częściowych reguł asocjacyjnych

W tej części rozdziału, na podstawie pracy [39], rozważane są algorytmy zachłanne z wagami dla konstruowania reguł asocjacyjnych. Każdy atrybut ma przypisaną wagę i należy zminimalizować całkowitą wagę atrybutów zawartych w częściowych regułach asocjacyjnych. Jeśli wagi atrybutów charakteryzują złożoność czasową obliczania wartości atrybutu, wówczas należy zminimalizować całkowitą złożoność czasową obliczania wartości atrybutów zawartych w częściowych regułach asocjacyjnych. Jeśli wagi atrybutów charakteryzują stopień ryzyka lub koszt pomiaru wartości atrybutu (np. w medycznych lub technicznych diagnozach), wówczas należy zminimalizować całkowite ryzyko (całkowity koszt).

Algorytm zachłanny z dwoma progami

Niech I będzie systemem informacyjnym zawierającym n wierszy i m kolumn oznaczonych przez atrybuty f_1, \dots, f_m , w będzie funkcją wagi dla I , która przypisuje dla każdego atrybutu f_i liczbę naturalną $w(f_i)$. Liczba $\sum_{j=1}^t w(f_{i_j})$ jest nazywana wagą reguły asocjacyjnej $(f_{i_1} = b_{i_1}) \wedge \dots \wedge (f_{i_t} = b_{i_t}) \rightarrow f_p = b_p$. Waga reguły, której lewa strona jest pusta, jest równa 0.

Niech α, γ będą liczbami rzeczywistymi takimi, że $0 \leq \gamma \leq \alpha < 1$. Algorytm 7 przedstawia algorytm zachłanny z progami α i γ , który konstruuje α -regułę asocjacyjną dla (I, r, f_p) i funkcji wagi w .

Algorytm 7: Algorytm zachłanny z dwoma progami α i γ dla konstruowania częściowej reguły asocjacyjnej

Wejście: System informacyjny I zawierający atrybuty f_1, \dots, f_m , wiersz $r = (b_1, \dots, b_m)$ z I , atrybut f_p z I , funkcja wagi w dla I , liczby rzeczywiste α i γ takie, że $0 \leq \gamma \leq \alpha < 1$.

Wyjście: α -reguła asocjacyjna dla (I, r, f_p) .

$Q \leftarrow \emptyset$; $D \leftarrow \emptyset$; $M \leftarrow \lceil |U(I, r, f_p)|(1 - \alpha) \rceil$; $N \leftarrow \lceil |U(I, r, f_p)|(1 - \gamma) \rceil$;

while $|D| < M$ **do**

wybierz $f_i \in \{f_1, \dots, f_m\} \setminus \{f_p\}$ o minimalnym indeksie i takim, że $U(I, r, f_p, f_i) \setminus D \neq \emptyset$ oraz wartość $\frac{w(f_i)}{\min\{|U(I, r, f_p, f_i) \setminus D|, N - |D|\}}$ jest minimalna;

$Q \leftarrow Q \cup \{f_i\}$; $D \leftarrow D \cup U(I, r, f_p, f_i)$;

end

return $\bigwedge_{f_i \in Q} (f_i = b_i) \rightarrow f_p = b_p$;

Przez $L_{\text{greedy}}^\gamma(\alpha) = L_{\text{greedy}}^\gamma(\alpha, I, r, f_p, w)$ jest oznaczana waga α -reguły asocjacyjnej konstruowanej przez rozważany algorytm dla danego systemu informacyjnego I , wiersza r , atrybutu f_p i funkcji wagi w . Algorytm 7 z równymi

progami α i $\gamma = \alpha$ jest nazywany *standardowym algorytmem zachłannym*. Przez $L_{\min}(\alpha) = L_{\min}(\alpha, I, r, f_p, w)$ jest oznaczana minimalna waga α -reguły asocjacyjnej dla I, r i f_p .

Dokładność standardowego algorytmu zachłannego

Dla naturalnej liczby m , oznaczamy $H(m) = 1 + \frac{1}{2} + \dots + \frac{1}{m}$. Wiadomo, że $\ln m \leq H(m) \leq \ln m + 1$.

Twierdzenie 47. *Niech I będzie systemem informacyjnym, r będzie wierszem z I , f_p będzie atrybutem z I , $U(I, r, f_p) \neq \emptyset$, w będzie funkcją wagi dla I , α będzie liczbą rzeczywistą taką, że $0 \leq \alpha < 1$. Wówczas $L_{\text{greedy}}^\alpha(\alpha) \leq L_{\min}(\alpha)H(|U(I, r, f_p)|)$.*

Twierdzenie 48. *Niech I będzie systemem informacyjnym zawierającym m kolumn oznaczonych przez atrybuty f_1, \dots, f_m , r będzie wierszem z I , f_p będzie atrybutem z I , $U(I, r, f_p) \neq \emptyset$, w będzie funkcją wagi dla I , α będzie liczbą rzeczywistą, $0 \leq \alpha < 1$. Wówczas $L_{\text{greedy}}^\alpha(\alpha) \leq L_{\min}(\alpha)H(\max_{i \in \{1, \dots, m\} \setminus \{p\}} |U(I, r, f_i)|)$.*

Wielomianowe algorytmy aproksymacyjne

Niech $0 \leq \alpha < 1$. Rozważany jest następujący problem: dla danego systemu informacyjnego I , wiersza r z I , atrybutu f_p z I , i funkcji wagi w dla I , należy znaleźć α -regułę asocjacyjną dla I, r i f_p o minimalnej wadze.

Twierdzenie 49. *Niech $0 \leq \alpha < 1$. Problem konstruowania α -reguły asocjacyjnej o minimalnej wadze jest NP-trudny.*

Twierdzenie 50. *Niech $0 \leq \alpha < 1$. Jeśli $NP \not\subseteq DTIME(n^{O(\log \log n)})$, wówczas dla dowolnego ε , $0 < \varepsilon < 1$, nie istnieje algorytm o złożoności wielomianowej, który dla danego systemu informacyjnego I , wiersza r z I , atrybutu f_p z I takiego, że $U(I, r, f_p) \neq \emptyset$, i funkcji wagi w dla I , konstruuje α -regułę asocjacyjną dla (I, r, f_p) , której waga wynosi najwyżej $(1-\varepsilon)L_{\min}(\alpha, I, r, f_p, w) \ln |U(I, r, f_p)|$.*

Algorytm zachłanny z dwoma progami vs. standardowy algorytm zachłanny

Niech I będzie binarnym systemem informacyjnym zawierającym n wierszy i m kolumn oznaczonych przez atrybuty f_1, \dots, f_m . Niech $r = (b_1, \dots, b_m)$ będzie wierszem z I i f_p będzie atrybutem z I . Powiemy, że trójka (I, r, f_p) jest *mniejszą trójką nad I* jeśli liczb $\neg b_p$ w kolumnie f_p jest co najmniej $\frac{n}{2}$.

Twierdzenie 51. *Niech będą badane binarne systemy informacyjne zawierające $n \geq 8$ wierszy i m kolumn oznaczonych przez atrybuty f_1, \dots, f_m . Część systemów informacyjnych I , w których dla każdej mniejszej trójki (I, r, f_p) nad I , istnieje funkcja wagi w dla I oraz liczby rzeczywiste α, γ takie, że $0 \leq \gamma < \alpha < 1$ i $L_{\text{greedy}}^\gamma(\alpha, I, r, f_p, w) < L_{\text{greedy}}^\alpha(\alpha, I, r, f_p, w)$, wynosi co najmniej*

$$1 - \frac{n^3 m 2^{4m}}{n^{\frac{m}{2}}} - \frac{n^2 m}{2^m}.$$

Rozszerzone algorytmy zachłanne

Przedstawione wyniki pokazują, że algorytm zachłanny z dwoma progami α i γ jest, w pewnym sensie, interesujący. Poniżej zostanie przedstawiona modyfikacja tego algorytmu, która pozwala wykorzystać zalety algorytmu zachłannego z dwoma progami pomijając badanie wszystkich wartości drugiego progu γ . Modyfikacja ta jest określana jako *rozszerzony* algorytm zachłanny.

Niech I będzie systemem informacyjnym zawierającym m kolumn oznaczonych przez atrybuty f_1, \dots, f_m , $r = (b_1, \dots, b_m)$ będzie wierszem z I , f_p będzie atrybutem z I , $U(I, r, f_p) \neq \emptyset$, w będzie funkcją wagi dla I , α będzie liczbą rzeczywistą taką, że $0 \leq \alpha < 1$.

Nie jest możliwe efektywnie badać wszystkie γ takie, że $0 \leq \gamma \leq \alpha$. Zamiast tego badane są wszystkie liczby naturalne N takie, że $M \leq N \leq |U(I, r, f_p)|$, gdzie $M = \lceil |U(I, r, f_p)|(1 - \alpha) \rceil$ (zobacz Algorytm 7). Dla każdego $N \in \{M, \dots, |U(I, r, f_p)|\}$, stosowany jest Algorytm 7 z parametrami M i N dla (I, r, f_p) i w , a następnie, spośród skonstruowanych α -reguł asocjacyjnych, wybierana jest α -reguła asocjacyjna o minimalnej wadze.

Granice $L_{\min}(\alpha)$ i $L_{\text{greedy}}^\gamma(\alpha)$

Na podstawie danych uzyskanych podczas pracy algorytmu zachłannego z dwoma progami została znaleziona najlepsza dolna granica wartości $L_{\min}(\alpha)$, zależna od tych danych. Algorytm zachłanny z progami α i γ jest stosowany dla systemu informacyjnego I , wiersza r , atrybutu f_p i funkcji wagi w . Podczas konstruowania α -reguły asocjacyjnej algorytm zachłanny sekwencyjnie wybiera atrybuty f_{g_1}, \dots, f_{g_t} .

Oznaczamy $U(I, r, f_p, f_{g_0}) = \emptyset$ i $\delta_0 = 0$. Dla $i = 1, \dots, t$, oznaczamy $\delta_i = |U(I, r, f_p, f_{g_i}) \setminus (U(I, r, f_p, f_{g_0}) \cup \dots \cup U(I, r, f_p, f_{g_{i-1}}))|$ i $w_i = w(f_{g_i})$. Jako dane o pracy algorytmu zachłannego stosowane są liczby $M_L = M_L(\alpha, \gamma, I, r, f_p, w) = \lceil |U(I, r, f_p)|(1 - \alpha) \rceil$ i $N_L = N_L(\alpha, \gamma, I, r, f_p, w) = \lceil |U(I, r, f_p)|(1 - \gamma) \rceil$, n-tki $\Delta_L = \Delta_L(\alpha, \gamma, I, r, f_p, w) = (\delta_1, \dots, \delta_t)$ i $W_L = W_L(\alpha, \gamma, I, r, f_p, w) = (w_1, \dots, w_t)$.

Dla $i = 0, \dots, t - 1$, oznaczamy $\varrho_i = \left\lceil \frac{w_{i+1}(M_L - (\delta_0 + \dots + \delta_i))}{\min\{\delta_{i+1}, N_L - (\delta_0 + \dots + \delta_i)\}} \right\rceil$. Parametr $\varrho_L(\alpha, \gamma) = \varrho_L(\alpha, \gamma, I, r, f_p, w)$ jest definiowany następująco:

$$\varrho_L(\alpha, \gamma) = \max \{ \varrho_i : i = 0, \dots, t - 1 \}.$$

Twierdzenie 52. *Parametr $\varrho_L(\alpha, \gamma)$ jest najlepszą dolną granicą $L_{\min}(\alpha)$ zależną od M_L , N_L , Δ_L , i W_L .*

Zbadano także pewne właściwości parametru $\varrho_L(\alpha, \gamma)$ i uzyskano dwie górne granice wartości $L_{\text{greedy}}^\gamma(\alpha)$, które nie zależą bezpośrednio od liczebności zbioru $U(I, r, f_p)$ i liczebności podzbiorów $U(I, r, f_p, f_i)$. Poniżej przedstawiono jedną z tych granic.

Twierdzenie 53. *Niech I będzie systemem informacyjnym, r będzie wierszem z I , f_p będzie atrybutem z I , $U(I, r, f_p) \neq \emptyset$, w będzie funkcją wagi dla I , α, γ będą liczbami rzeczywistymi i $0 \leq \gamma < \alpha < 1$. Wówczas $L_{\text{greedy}}^\gamma(\alpha, I, r, f_p, w) < \varrho_L(\gamma, \gamma, I, r, f_p, w) \left(\ln \left(\frac{1-\gamma}{\alpha-\gamma} \right) + 1 \right)$.*

Wniosek 7. Niech ε będzie liczbą rzeczywistą oraz $0 < \varepsilon < 1$. Wówczas dla dowolnego α , $\varepsilon \leq \alpha < 1$, zachodzi: $\varrho_L(\alpha, \alpha) \leq L_{\min}(\alpha) \leq L_{\text{greedy}}^{\alpha-\varepsilon}(\alpha) < \varrho_L(\alpha - \varepsilon, \alpha - \varepsilon) \left(\ln\left(\frac{1}{\varepsilon}\right) + 1\right)$.

Na przykład, $\ln\left(\frac{1}{0.01}\right) + 1 < 5.61$ i $\ln\left(\frac{1}{0.1}\right) + 1 < 3.31$. Uzyskane wyniki pokazują, że dolna granica $L_{\min}(\alpha) \geq \varrho_L(\alpha, \alpha)$ nie jest trywialna.

7.3 Podejście oparte na dynamicznym programowaniu dla optymalizacji reguł asocjacyjnych

W tym rozdziale, na podstawie pracy [67], przedstawiono zastosowanie podejścia opartego na dynamicznym programowaniu dla optymalizacji reguł asocjacyjnych względem pokrycia. Rozważane podejście może zostać w prosty sposób rozszerzone dla innych funkcji kosztu np. długość, liczba błędnych zaklasyfikowań oraz dla sekwencyjnej optymalizacji względem wybranych funkcji kosztu.

Rozdział 6 poświęcony był badaniu podejścia opartego na dynamicznym programowaniu dla optymalizacji reguł decyzyjnych a rozważane tablice decyzyjne były spójne tzn. nie zawierały wierszy o tych samych wartościach atrybutów warunkowych i różnych decyzjach. W sytuacji, kiedy badane są reguły asocjacyjne dla systemów informacyjnych i kiedy każdy atrybut jest sekwencyjnie rozważany jako atrybut decyzyjny, mogą pojawić się niespójne tablice decyzyjne, tj. tablice zawierające równe wiersze o różnych decyzjach. Zatem podejście przedstawione w rozdziale 6.1, zostało rozszerzone dla przypadku niespójnych tablic decyzyjnych. Wymagało to wykonania zmian w definicjach, algorytmach (nowy warunek stopu), dowodach poprawności algorytmów oraz w oprogramowaniu.

Niech I będzie systemem informacyjnym zawierającym n wierszy i m kolumn oznaczonych przez atrybuty f_1, \dots, f_m . Dla każdego $f_p \in \{f_1, \dots, f_m\}$, system informacyjny I jest transformowany do postaci tablicy decyzyjnej I_{f_p} . Kolumna f_p jest usuwana z I i powstaje tabela z $m - 1$ kolumn oznaczonych przez atrybuty $f_1, \dots, f_{p-1}, f_{p+1}, \dots, f_m$. Wartości atrybutu f_p zostają dołączone do wierszy uzyskanej tabeli, która jest oznaczona jako I_{f_p} . Wartości te są interpretowane jako decyzje. Niech r będzie wierszem z I i r_{f_p} będzie wierszem z I_{f_p} uzyskanym z r przez usunięcie wartości atrybutu na przecięciu z kolumną f_p . Definicja nieredukowalnej 0-reguły asocjacyjnej dla (I, r, f_p) jest pominięta, gdyż jest bliska definicji nieredukowalnej reguły decyzyjnej dla I_{f_p} i r_{f_p} . Zamiast nieredukowalnych 0-reguł asocjacyjnych dla (I, r, f_p) , można badać nieredukowalne reguły decyzyjne dla I_{f_p} i r_{f_p} .

Podejście oparte na dynamicznym programowaniu dla badania tablic decyzyjnych I_{f_p} jest bliskie podejściu dynamicznego programowania rozważanemu w poprzednim rozdziale. Główną różnicą jest to, że I_{f_p} może zawierać równe wiersze o tych samych lub różnych decyzjach. W szczególności, należy rozważyć inną definicję pojęcia tabela zdegenerowana. Niespójna tablica decyzyjna T jest nazywana *zdegenerowana*, jeśli jest pusta lub wszystkie wiersze w T mają przypisaną tą samą decyzję lub wszystkie wiersze w T są równe. Zmodyfikowany został algorytm, który konstruuje skierowany graf acykliczny $\Delta_0(T)$ dla danej

Tabela 4: Maksymalne pokrycie reguł asocjacyjnych

Zbiór danych	Wiersze	Attr	Min	Avg	Max
adult-stretch	16	5	4	7.0	8
breast-cancer	366	10	5	41.143	65
cars	1728	7	1	336.319	576
hayes-roth	60	3	2	7.435	13
lenses	24	5	2	7.958	12
monks-1-test	432	7	12	45.0	108
monks-3-test	432	7	36	56.0	108
shuttle-landing	15	7	6	7.8	9
teeth	23	9	8	12.652	15
tic-tac-toe	958	10	12	67.282	90
zoo-data	59	17	31	34.814	35

tablicy decyzyjnej T oraz procedura optymalizacji reguł decyzyjnych względem pokrycia.

Niech r będzie wierszem z I . Dla dowolnego $f_p \in \{f_1, \dots, f_m\}$, 0-reguły asocjacyjne dla (I, r, f_p) będą nazywane regułami asocjacyjnymi dla I i r . Celem jest opisanie całego zbioru nieredukowalnych reguł asocjacyjnych dla I i r o maksymalnym pokryciu.

Dla każdego $f_p \in \{f_1, \dots, f_m\}$, konstruowana jest tablica decyzyjna I_{f_p} i graf $\Delta_0(I_{f_p})$. Dla wiersza r , tworzony jest wiersz r_{f_p} i zostaje zastosowana procedura optymalizacji względem pokrycia do grafu $\Delta_0(I_{f_p})$ i wiersza r_{f_p} . W wyniku można opisać cały zbiór $Rul_{\Delta_0^c(I_{f_p})}(I_{f_p}, r_{f_p})$ nieredukowalnych reguł decyzyjnych dla I_{f_p} i r_{f_p} o maksymalnym pokryciu $Opt_{\Delta_0(I_{f_p})}^c(I_{f_p}, r_{f_p})$. Niech $c(I, r) = \max\{Opt_{\Delta_0(I_{f_p})}^c(I_{f_p}, r_{f_p}) : p = 1, \dots, m\}$. Wówczas zbiór

$$\bigcup_{f_p \in \{f_1, \dots, f_m\}, Opt_{\Delta_0(I_{f_p})}^c(I_{f_p}, r_{f_p}) = c(I, r)} Rul_{\Delta_0^c(I_{f_p})}(I_{f_p}, r_{f_p})$$

jest równy zbiorowi wszystkich nieredukowalnych reguł asocjacyjnych o maksymalnym pokryciu dla I i r .

Tabela 4 przedstawia wyniki eksperymentów związane z maksymalnym pokryciem nieredukowalnych reguł asocjacyjnych. Każdy zbiór danych z Repozytorium Uczenia Maszynowego [11], traktowany był jako system informacyjny I . Dla każdego wiersza r z I , zostało uzyskane maksymalne pokrycie nieredukowalnej reguły asocjacyjnej dla I i r . Następnie, dla wierszy z I została wyliczona minimalna wartość pokrycia reguły asocjacyjnej dla I i r o maksymalnym pokryciu (kolumna "Min"), maksymalna wartość pokrycia takiej reguły (kolumna "Max") i średnia wartość pokrycia reguł asocjacyjnych o maksymalnym pokryciu – jedna reguła dla każdego wiersza (kolumna "Avg").

W pracy [67] przedstawiono także wyniki dotyczące średniej liczby wierzchołków i krawędzi dla grafów $\Delta_0(I_{f_p})$, $p = 1, \dots, m$. Dla rozważanych, małych tablic decyzyjnych, uzyskane wyniki są rozsądne. Dla dużych tablic decyzyjnych należy rozważać nie cały graf tylko jego część – jak to zostało przedstawione w rozdziale 6.2.

7.4 Podejście oparte na dynamicznym programowaniu dla konstruowania systemów reguł asocjacyjnych

W tym rozdziale, na podstawie pracy [4], zostanie opisany algorytm, który konstruuje, dla danego systemu informacyjnego, system reguł asocjacyjnych, które pokrywają większą część wierszy. Dla większości rozważanych systemów informacyjnych liczebność konstruowanego systemu reguł jest rozsądna. Rozważany algorytm symuluje, w pewnym sensie, pracę algorytmu zachłannego dla problemu pokrycia zbioru: algorytm w każdej iteracji wybiera regułę, która pokrywa maksymalną liczbę wierszy wcześniej niepokrytych. Fakt ten pozwolił uzyskać nietrywialne dolne granice minimalnej liczebności systemu reguł, na podstawie danych uzyskanych podczas pracy tego algorytmu. Rozważany algorytm używa metody dynamicznego programowania do wyboru reguły, która pokrywa maksymalną liczbę niepokrytych wcześniej wierszy.

Podstawowe pojęcia

System informacyjny I jest tabelą posiadającą $n + 1$ kolumn oznaczonych przez atrybuty f_1, \dots, f_{n+1} . Wiersze tabeli są wypełnione przez nieujemne liczby całkowite interpretowane jako wartości atrybutów.

Regułą asocjacyjną dla I jest reguła postaci

$$(f_{i_1} = a_1) \wedge \dots \wedge (f_{i_m} = a_m) \rightarrow f_j = a,$$

gdzie $f_j \in \{f_1, \dots, f_{n+1}\}$, $f_{i_1}, \dots, f_{i_m} \in \{f_1, \dots, f_{n+1}\} \setminus \{f_j\}$, i a, a_1, \dots, a_m są nieujemnymi liczbami całkowitymi.

Pojęcie reguły asocjacyjnej dla I jest związane z pojęciem tablicy decyzyjnej i reguły decyzyjnej. Badane są dwa rodzaje tablic decyzyjnych: tablice z wielowartościową decyzją oraz tablice z jednowartościową decyzją.

Tablica decyzyjna z wielowartościową decyzją T jest tabelą posiadającą n kolumn oznaczonych przez atrybuty warunkowe f_1, \dots, f_n . Wiersze tej tabeli są parami różne i są wypełnione przez nieujemne liczby całkowite interpretowane jako wartości atrybutów warunkowych. Każdy wiersz r ma przypisany skończony, niepusty zbiór $D(r)$ nieujemnych liczb całkowitych interpretowanych jako decyzje (wartości atrybutu decyzyjnego). Dla danego wiersza r z T , należy znaleźć decyzję ze zbioru $D(r)$.

Tablica decyzyjna z jednowartościową decyzją T jest tabelą posiadającą n kolumn oznaczonych przez atrybuty warunkowe f_1, \dots, f_n . Wiersze tej tabeli są parami różne i są wypełnione przez nieujemne liczby całkowite interpretowane jako wartości atrybutów warunkowych. Każdy wiersz r ma przypisaną nieujemną liczbę całkowitą $d(r)$ interpretowaną jako wartość atrybutu decyzyjnego. Dla danego wiersza r z T , należy znaleźć decyzję $d(r)$. Tablice decyzyjne z jednowartościową decyzją mogą być traktowane jako specjalny rodzaj tablic decyzyjnych z wielowartościową decyzją, w których $D(r) = \{d(r)\}$ dla każdego wiersza r .

Dla każdego atrybutu $f_i \in \{f_1, \dots, f_{n+1}\}$, system informacyjny I jest przekształcany do postaci tablicy decyzyjnej I_{f_i} . Kolumna f_i zostaje usunięta z I , powstaje tabela zawierająca n kolumn oznaczonych przez atrybuty $f_1, \dots, f_{i-1}, f_{i+1}, \dots, f_{n+1}$.

Wartości atrybutu f_i zostają przypisane do wierszy uzyskanej tabeli I_{f_i} jako decyzje.

Tabela I_{f_i} może zawierać równe wiersze. Tabela ta jest przekształcana do postaci dwóch tablic decyzyjnych: z wielowartościową decyzją i z jednowartościową decyzją. Tablica decyzyjna $I_{f_i}^{m-v}$ z wielowartościową decyzją powstaje z tabeli I_{f_i} przez zastąpienie każdej grupy równych wierszy, jednym wierszem z grupy do którego zostaje przypisany zbiór decyzji, które były przypisane do wszystkich wierszy z grupy. Tablica decyzyjna $I_{f_i}^{s-v}$ z jednowartościową decyzją powstaje z tabeli I_{f_i} przez zastąpienie każdej grupy równych wierszy, jednym wierszem z grupy do którego zostaje przypisana najbardziej wspólna decyzja dla grupy.

Przez $\Phi^{m-v}(I)$ jest oznaczany zbiór $\{I_{f_1}^{m-v}, \dots, I_{f_{n+1}}^{m-v}\}$ tablic decyzyjnych z wielowartościową decyzją uzyskanych z systemu informacyjnego I . Przez $\Phi^{s-v}(I)$ jest oznaczany zbiór $\{I_{f_1}^{s-v}, \dots, I_{f_{n+1}}^{s-v}\}$ tablic decyzyjnych z jednowartościową decyzją, uzyskanych z systemu informacyjnego I . Ponieważ tablice decyzyjne z jednowartościową decyzją są szczególnym przypadkiem tablic decyzyjnych z wielowartościową decyzją, przedstawione zostanie pojęcie reguły decyzyjnej dla tablic z wielowartościową decyzją.

Niech $T \in \Phi^{m-v}(I)$. Dla uproszczenia, niech $T = I_{f_{n+1}}^{m-v}$. Atrybut f_{n+1} jest rozważany jako atrybut decyzyjny tabeli T . Przez $Row(T)$ jest oznaczany zbiór wierszy z T . Niech $D(T) = \bigcup_{r \in Row(T)} D(r)$.

Tablica T jest nazywana *zdegenerowana* jeśli jest pusta lub posiada *wspólną* decyzję, tzn. $\bigcap_{r \in Row(T)} D(r) \neq \emptyset$. Przez $N(T)$ jest oznaczana liczba wierszy w tabeli T i dla dowolnego $t \in \omega$, gdzie ω jest zbiorem nieujemnych liczb całkowitych, przez $N(T, t)$ jest oznaczana liczba wierszy r z T takich, że $t \in D(r)$. Przez $mcd(T)$ jest oznaczana *najbardziej wspólna decyzja* dla T , która jest minimalną decyzją t_0 z $D(T)$ taką, że $N(T, t_0) = \max\{N(T, t) : t \in D(T)\}$. Jeśli T jest pusta wówczas $mcd(T) = 0$.

Dla dowolnego atrybutu warunkowego $f_i \in \{f_1, \dots, f_n\}$, przez $E(T, f_i)$ jest oznaczany zbiór wartości atrybutu f_i z tabeli T . Przez $E(T)$ jest oznaczany zbiór atrybutów warunkowych dla których $|E(T, f_i)| \geq 2$.

Niech T będzie niepustą tablicą decyzyjną. *Podtabela* tabeli T to tabela uzyskana z T przez usunięcie pewnych wierszy. Niech $f_{i_1}, \dots, f_{i_m} \in \{f_1, \dots, f_n\}$ i $a_1, \dots, a_m \in \omega$. Przez $T(f_{i_1}, a_1) \dots (f_{i_m}, a_m)$ jest oznaczana podtabela tabeli T zawierająca wiersze z T , które na przecięciu z kolumnami f_{i_1}, \dots, f_{i_m} posiadają, odpowiednio, wartości a_1, \dots, a_m .

Jako *miara niepewności* dla niepustej tablicy decyzyjnej badana jest *względna liczba błędnych zaklasyfikowań* $rme(T) = (N(T) - N(T, mcd(T)))/N(T)$, gdzie $N(T, mcd(T))$ jest liczbą wierszy r w T zawierających najbardziej wspólną decyzję dla T w $D(r)$.

Regułą decyzyjną nad T jest wyrażenie postaci

$$(f_{i_1} = a_1) \wedge \dots \wedge (f_{i_m} = a_m) \rightarrow f_{n+1} = t \quad (9)$$

gdzie $f_{i_1}, \dots, f_{i_m} \in \{f_1, \dots, f_n\}$ i a_1, \dots, a_m, t są liczbami ze zbioru ω . Możliwe jest, że $m = 0$. Dla rozważanej reguły, $T^0 = T$ i jeśli $m > 0$, dla $j = 1, \dots, m$,

$T^j = T(f_{i_1}, a_1) \dots (f_{i_j}, a_j)$. Reguła decyzyjna (9) pokrywa wiersz $r = (b_1, \dots, b_n)$ z T , jeśli r należy do T^m , tj. $b_{i_1} = a_1, \dots, b_{i_m} = a_m$.

Reguła decyzyjna (9) nad T jest nazywana *regułą decyzyjną dla T* , jeśli $t = \text{mcd}(T^m)$ i jeśli $m > 0$, wówczas T^{j-1} jest niezdegenerowana dla $j = 1, \dots, m$ i $f_{i_j} \in E(T^{j-1})$. Przez $DR(T)$ jest oznaczany zbiór reguł decyzyjnych dla T .

Niech ρ będzie regułą decyzyjną dla T , która jest równa (9). Wartość $\text{rme}(T, \rho) = \text{rme}(T^m)$ jest nazywana *niepewnością* reguły ρ . Niech α będzie liczbą rzeczywistą taką, że $0 \leq \alpha \leq 1$. Reguła decyzyjna ρ dla T jest *α -regułą decyzyjną dla T* , jeśli $\text{rme}(T, \rho) \leq \alpha$. Jeśli $\alpha = 0$ (w tym przypadku, dla każdego wiersza r pokrytego przez ρ , zbiór $D(r)$ zawiera decyzję z prawej strony reguły ρ), wówczas ρ jest *dokładną* regułą. Przez $DR_\alpha(T)$ jest oznaczany zbiór α -reguł decyzyjnych dla T .

Algorytm dla konstruowania systemu reguł asocjacyjnych

α -Reguły decyzyjne dla tablic z $\Phi^{m-v}(I)$ mogą być rozważane jako α -reguły asocjacyjne (modyfikacja dla modelu tablic decyzyjnych z wielowartościową decyzją) dla systemu informacyjnego I . α -Reguły decyzyjne dla tablic decyzyjnych z $\Phi^{s-v}(I)$ mogą być rozważane jako α -reguły asocjacyjne (modyfikacja dla modelu tablic decyzyjnych z jednowartościową decyzją) dla systemu informacyjnego I . Poniżej zostanie przedstawiony algorytm dla konstruowania systemu reguł asocjacyjnych dla I zarówno w ramach modelu tablic decyzyjnych z wielowartościową decyzją, jak i modelu tablic decyzyjnych z jednowartościową decyzją. Ponieważ tablice decyzyjne z jednowartościową decyzją mogą być traktowane jako specjalny rodzaj tablic decyzyjnych z wielowartościową decyzją, przedstawione wyniki dotyczą głównie modelu tablic decyzyjnych z wielowartościową decyzją.

Niech $T = I_{f_{n+1}}^{m-v}$. Niech S będzie niepustym, skończonym zbiorem α -reguł decyzyjnych dla T (*system α -reguł decyzyjnych dla T*) i β będzie liczbą rzeczywistą taką, że $0 \leq \beta \leq 1$. S jest nazywany *β -systemem α -reguł decyzyjnych dla T* , jeśli reguły z S pokrywają co najmniej $(1 - \beta)N(T)$ wierszy z T .

Badany jest algorytm α - β -Reguły, który dla tablicy decyzyjnej T oraz liczb rzeczywistych α i β , $0 \leq \alpha \leq 1$ i $0 \leq \beta \leq 1$, konstruuje β -system α -reguł decyzyjnych dla T . W każdym kroku algorytm wybiera (korzystając z algorytmu dynamicznego programowania [3]) regułę decyzyjną, która pokrywa maksymalną liczbę niepokrytych dotychczas wierszy. Algorytm kończy pracę, kiedy skonstruowane reguły pokrywają co najmniej $(1 - \beta)N(T)$ wierszy z T . Przez $Rule_{\alpha, \beta}(T)$ jest oznaczany skonstruowany system reguł.

Przez $C(T, \alpha, \beta)$ jest oznaczana minimalna liczebność β -systemu α -reguł decyzyjnych dla T . Oczywiście jest, że $C(T, \alpha, \beta) \leq |Rule_{\alpha, \beta}(T)|$. Korzystając z danych uzyskanych podczas pracy algorytmu α - β -Reguły, można uzyskać dolną granicę parametru $C(T, \alpha, \beta)$. Niech podczas konstruowania β -systemu α -reguł decyzyjnych dla T , algorytm α - β -Reguły wybiera odpowiednio reguły ρ_1, \dots, ρ_t . Niech B_1, \dots, B_t będą zbiorami wierszy z T pokrytymi, odpowiednio, przez reguły ρ_1, \dots, ρ_t . Zbiór $B_0 = \emptyset$, $\delta_0 = 0$ i, dla $i = 1, \dots, t$, $\delta_i = |B_i \setminus (B_0 \cup \dots \cup B_{i-1})|$. Dane uzyskane podczas pracy algorytmu to n -tka $(\delta_1, \dots, \delta_t)$ oraz liczby $N(T)$ i β .

Na podstawie wyników uzyskanych w [38] dotyczących pracy algorytmu zachłannego dla problemu pokrycia zbioru wynika, że $C(T, \alpha, \beta) \geq l(T, \alpha, \beta)$, gdzie

$$l(T, \alpha, \beta) = \max \left\{ \left\lceil \frac{[(1 - \beta)N(T)] - (\delta_0 + \dots + \delta_i)}{\delta_{i+1}} \right\rceil : i = 0, \dots, t - 1 \right\}.$$

W wyniku zastosowania algorytmu α - β -Reguły, dla każdej tablicy decyzyjnej $T \in \Phi^{m-v}(I)$, konstruowany jest zbiór reguł $Rule_{\alpha, \beta}(T)$. W wyniku powstaje system reguł (α -reguły asocjacyjne dla systemu informacyjnego I – modyfikacja dla modelu tablic decyzyjnych z wielowartościową decyzją) $Rule_{\alpha, \beta}^{m-v}(I) = \bigcup_{T \in \Phi^{m-v}(I)} Rule_{\alpha, \beta}(T)$. System ten zawiera, dla każdego $T \in \Phi^{m-v}(I)$, podsystem, który jest β -systemem α -reguł decyzyjnych dla T . Przez $C^{m-v}(I, \alpha, \beta)$ jest oznaczana minimalna liczebność tego systemu. Można pokazać, że

$$L^{m-v}(I, \alpha, \beta) \leq C^{m-v}(I, \alpha, \beta) \leq U^{m-v}(I, \alpha, \beta),$$

$$L^{m-v}(I, \alpha, \beta) = \sum_{T \in \Phi^{m-v}(I)} l(T, \alpha, \beta) \text{ i } U^{m-v}(I, \alpha, \beta) = \left| Rule_{\alpha, \beta}^{m-v}(I) \right|.$$

Takie samo podejście można zastosować dla zbioru $\Phi^{s-v}(I)$ tablic decyzyjnych z jednowartościową decyzją. Wynikiem jest system reguł (α -reguły asocjacyjne dla systemu informacyjnego I – modyfikacja dla modelu tablic decyzyjnych z jednowartościową decyzją) $Rule_{\alpha, \beta}^{s-v}(I) = \bigcup_{T \in \Phi^{s-v}(I)} Rule_{\alpha, \beta}(T)$, który zawiera dla każdego $T \in \Phi^{s-v}(I)$, podsystem, który jest β -systemem α -reguł decyzyjnych dla T . Przez $C^{s-v}(I, \alpha, \beta)$ jest oznaczana minimalna liczebność takiego systemu. Można pokazać, że

$$L^{s-v}(I, \alpha, \beta) \leq C^{s-v}(I, \alpha, \beta) \leq U^{s-v}(I, \alpha, \beta),$$

$$L^{s-v}(I, \alpha, \beta) = \sum_{T \in \Phi^{s-v}(I)} l(T, \alpha, \beta) \text{ i } U^{s-v}(I, \alpha, \beta) = \left| Rule_{\alpha, \beta}^{s-v}(I) \right|.$$

Wyniki eksperymentów

Eksperymenty zostały wykonane na zbiorach umieszczonych w Repozytorium Uczenia Maszynowego [11]. Przygotowano 12 zbiorów danych, które były traktowane jako systemy informacyjne. Tabela 5 przedstawia górne/dolne granice wartości $C^{m-v}(I, \alpha, \beta)$ (kolumna “many-val”) i dla $C^{s-v}(I, \alpha, \beta)$ (kolumna “single-val”), dla par $(\alpha = 0, \beta = 0)$ i $(\alpha = 0.3, \beta = 0.2)$.

Można zauważyć, dla tablic z wielowartościową decyzją, że górne i dolne granice liczby reguł są mniejsze lub równe wartościom granic dla tablic decyzyjnych z jednowartościową decyzją. Podczas eksperymentów ustawiono próg $30 \times (\text{liczba atrybutów})$ jako rozsądną górną granicę liczby reguł, jeśli system reguł stosowany jest dla reprezentacji wiedzy. W przypadku $\alpha = 0$ i $\beta = 0$, próg został przekroczony dla pięciu systemów informacyjnych (wartości pogrubione): balance-scale, breast-cancer, cars, hayes-roth-data i tic-tac-toe. Badanie przybliżonych reguł oraz częściowych pokryć może poprawić tę sytuację. W przypadku $\alpha = 0.3$ i $\beta = 0.2$, próg został przekroczony dla trzech systemów informacyjnych:

Tabela 5: Całkowita liczba reguł (górną granicą / dolną granicą)

System informacyjny	Wiersze	Attr	$\alpha = 0, \beta = 0$		$\alpha = 0.3, \beta = 0.2$	
			single-val	many-val	single-val	many-val
adult-stretch	16	5	11	9	11	9
balance-scale	625	5	847	507	390	50
breast-cancer	266	10	1027	984	401	362
cars	1728	7	448	358	62	51
hayes-roth-data	69	5	169	161	115	105
lenses	24	5	29	25	17	15
monks-1-test	432	7	50	40	35	31
monks-3-test	432	7	36	22	25	16
shuttle-landing	15	7	47	42	30	27
teeth	23	9	114	114	65	65
tic-tac-toe	958	10	5186	5186	1508	1508
zoo-data	59	17	159	143	51	49

balance-scale, breast-cancer i tic-tac-toe, kiedy badane były tablice decyzyjne z jednowartościową decyzją i dla dwóch systemów informacyjnych: breast-cancer i tic-tac-toe, kiedy badane były tablice decyzyjne z wielowartościową decyzją.

W przyszłych badaniach przedstawione podejście zostanie rozszerzone o podejście zaproponowane w [65] (rozdział 6.2) dla konstruowania systemów reguł asocjacyjnych. Podejście to pozwala konstruować reguły o wartościach pokrycia bliskich wartościom maksymalnym i wymaga mniej czasu, niż podejście oparte na dynamicznym programowaniu.

7.5 Porównanie heurystyk dla optymalizacji reguł asocjacyjnych

W rozdziale tym, na podstawie pracy [5], zostanie przedstawionych pięć heurystyk zachłanych dla konstruowania reguł asocjacyjnych i zostaną one porównane z punktu widzenia długości i pokrycia konstruowanych reguł. Uzyskane wyniki zostaną także porównane z wynikami optymalnymi dotyczącymi pokrycia i długości reguł decyzyjnych konstruowanych za pomocą algorytmów dynamicznego programowania. Średnia relatywna różnica pomiędzy długością konstruowanych reguł przez najlepszą heurystykę a minimalną długością reguł wynosi najwyżej 4%. Taka sama sytuacja występuje w przypadku pokrycia.

Poza nowymi pojęciami zostały wykorzystane definicje i pojęcia z rozdziału 7.4. Niech I będzie systemem informacyjnym zawierającym $n + 1$ kolumn oznaczonych przez atrybuty f_1, \dots, f_{n+1} . Powiązany jest z nim zbiór $\Phi^{m-v}(I) = \{I_{f_1}^{m-v}, \dots, I_{f_{n+1}}^{m-v}\}$ tablic decyzyjnych z wielowartościową decyzją i zbiór $\Phi^{s-v}(I) = \{I_{f_1}^{s-v}, \dots, I_{f_{n+1}}^{s-v}\}$ tablic decyzyjnych z jednowartościową decyzją. Zbiory te powstały przez transformację systemu informacyjnego I .

Ponieważ tablice decyzyjne z jednowartościową decyzją są szczególnym przypadkiem tablic decyzyjnych z wielowartościową decyzją, rozważany będzie model tablic decyzyjnych z wielowartościową decyzją. Niech $T \in \Phi^{m-v}(I)$. Dla

uproszczenia, niech $T = I_{f_{n+1}}^{m-v}$. Atrybut f_{n+1} będzie rozważany jako atrybut decyzyjny tabeli T . Przez $N(T)$ jest oznaczana liczba wierszy w tabeli T . Dla decyzji a , przez $N(T, a)$ jest oznaczana liczba wierszy r z T takich, że $a \in D(r)$, i $M(T, a) = N(T) - N(T, a)$. Decyzja a jest *wspólną* decyzją dla T jeśli $a \in D(r)$ dla dowolnego wiersza r z T . Niech r będzie wierszem z T . Dokładna reguła decyzyjna dla T , która pokrywa wiersz r jest nazywana *regułą decyzyjną dla T i r* . Liczba wierszy pokrytych przez tą regułę jest nazywana *pokryciem* reguły. *Długość* reguły to liczba deskryptorów (par "atrybut=wartość") z lewej strony reguły.

Heurystyki zachłanne

W tej części rozdziału zostanie przedstawiona praca pięciu heurystyk zachłanych na przykładzie tabeli $T = I_{f_{n+1}}^{m-v}$. Niech $r = (b_1, \dots, b_n)$ będzie wierszem w T i a będzie decyzją z $D(r)$. Heurystka H konstruuje regułę decyzyjną dla T i r . Heurystyka na początku konstruuje regułę, której lewa strona jest pusta $\rightarrow f_{n+1} = a$, a następnie sekwencyjnie dodawane są deskryptory do lewej strony reguły. Niech podczas pracy heurystyki H , została skonstruowana reguła:

$$(f_{i_1} = b_{i_1}) \wedge \dots \wedge (f_{i_m} = b_{i_m}) \rightarrow f_{n+1} = a.$$

Regule tej odpowiada podtabela $T' = T(f_{i_1}, b_{i_1}), \dots, (f_{i_m}, b_{i_m})$ tabeli T . Jeśli a jest wspólną decyzją dla T' wówczas praca heurystyki H zostaje zakończona i wynikiem jest skonstruowana reguła. W przeciwnym przypadku, wybierany jest nowy atrybut $f_{i_{m+1}}$ i konstruowana jest nowa reguła:

$$(f_{i_1} = b_{i_1}) \wedge \dots \wedge (f_{i_m} = b_{i_m}) \wedge (f_{i_{m+1}} = b_{i_{m+1}}) \rightarrow f_{n+1} = a.$$

Przyjęte są oznaczenia $T'' = T'(f_{i_{m+1}}, b_{i_{m+1}})$, $M(f_{i_{m+1}}, r, a) = M(T'', a) = N(T'') - N(T'', a)$ i $RM(f_{i_{m+1}}, r, a) = (N(T'') - N(T'', a)) / N(T'')$, oraz $\alpha(f_{i_{m+1}}, r, a) = N(T', a) - N(T'', a)$ i $\beta(f_{i_{m+1}}, r, a) = M(T', a) - M(T'', a)$. Poniżej zostanie opisane jak pięć heurystyk zachłanych wybiera atrybut $f_{i_{m+1}}$.

Heurystyka "M" wybiera atrybut $f_{i_{m+1}} \in E(T')$, który minimalizuje wartość $M(f_{i_{m+1}}, r, a)$.

Heurystyka "RM" wybiera atrybut $f_{i_{m+1}} \in E(T')$, który minimalizuje wartość $RM(f_{i_{m+1}}, r, a)$.

Heurystyka "maxCov" wybiera atrybut $f_{i_{m+1}} \in E(T')$, który minimalizuje wartość $\alpha(f_{i_{m+1}}, r, a)$ biorąc po uwagę, że $\beta(f_{i_{m+1}}, r, a) > 0$.

Heurystyka "poly" wybiera atrybut $f_{i_{m+1}} \in E(T')$, który maksymalizuje wartość $\frac{\beta(f_{i_{m+1}}, r, a)}{\alpha(f_{i_{m+1}}, r, a) + 1}$.

Heurystyka "log" wybiera atrybut $f_{i_{m+1}} \in E(T')$, który maksymalizuje wartość $\frac{\beta(f_{i_{m+1}}, r, a)}{\log_2(\alpha(f_{i_{m+1}}, r, a) + 2)}$.

Niech H będzie jedną z rozważanych heurystyk. Dla wiersza r tabeli T , zostanie zastosowana heurystyka H do wiersza r i do każdej decyzji $a \in D(r)$. Wynikiem jest $|D(r)|$ reguł. W zależności od celu, albo spośród tych reguł wybierana jest reguła o minimalnej długości, albo reguła o maksymalnym pokryciu.

Wyniki eksperymentów

Eksperymenty zostały wykonane dla 12 systemów informacyjnych uzyskanych z Repozytorium Uczenia Maszynowego [11].

Dla każdego systemu informacyjnego I , został skonstruowany zbiór $\Phi^{m-v}(I)$ tablic decyzyjnych z wielowartościową decyzją oraz zbiór $\Phi^{s-v}(I)$ tablic decyzyjnych z jednowartościową decyzją. Dla każdego wiersza r każdej tabeli $T \in \Phi^{m-v}(I)$, została zastosowana każda z rozważanych pięciu heurystyk zachłanych, tak jak zostało to wcześniej opisane. Heurystykom zastosowanym do wiersza r zostały przypisane rangi odnośnie długości i pokrycia konstruowanych reguł oraz została znaleziona, dla każdej heurystyki, średnia ranga odnośnie długości i pokrycia spośród wszystkich wierszy dla wszystkich tabel $\Phi^{m-v}(I)$. Następnie została uzyskana średnia spośród średnich rang 12 systemów informacyjnych. Wyniki przedstawia Tabela 11. Najlepsze trzy heurystyki dla długości to M, log i RM. Najlepsze trzy heurystyki dla pokrycia to poly, log i RM. W ten sam sposób zostały zbadane tablice decyzyjne z jednowartościową decyzją (zobacz Tabela 11). Najlepsze trzy heurystyki dla długości to M, RM i log. Najlepsze trzy heurystyki dla pokrycia to poly, log, and RM.

Dla każdej heurystki i dla każdego wiersza r każdej tabeli $T \in \Phi^{m-v}(I)$, została porównana długość reguły konstruowanej przez heurystykę dla r (jest ona oznaczona jako *length_greedy*) z minimalną długością reguły (jest ona oznaczona jako *length_min*) i została obliczona relatywna różnica $\frac{\text{length_greedy} - \text{length_min}}{\text{length_min}}$ (zakładając, że $\frac{0}{0} = 0$). Minimalna długość reguły została wyznaczona z wykorzystaniem algorytmu dynamicznego programowania. Następnie została obliczona średnia relatywna różnica spośród wszystkich wierszy wszystkich tablic z $\Phi^{m-v}(I)$, i została uzyskana ogólna średnia relatywna różnica dla wszystkich 12 systemów informacyjnych. Wyniki przedstawia Tabela 12. Najlepsze trzy heurystyki dla długości to M (2% różnicy), RM (4%) i log (13%). Takie same badania zostały wykonane dla pokrycia i tablic decyzyjnych z wielowartościową decyzją. Średnia relatywna różnica została wyznaczona jako $\frac{\text{coverage_max} - \text{coverage_greedy}}{\text{coverage_max}}$, gdzie *coverage_greedy* jest pokryciem reguły konstruowanej przez heurystykę zachłanną i *coverage_max* jest maksymalnym pokryciem reguły obliczonym za pomocą algorytmu dynamicznego programowania. Najlepsze trzy heurystyki dla pokrycia to poly (4% różnicy), log (8%) i maxCov (14%).

Tabela 11: Ogólne rangi dla heurystyk

		Heuristics				
		poly	log	maxCov	M	RM
Single-valued decisions	Length	3.38	2.25	5.00	2.17	2.21
	Coverage	1.67	1.83	4.00	4.21	3.29
Many-valued decisions	Length	3.33	2.33	5.00	1.79	2.54
	Coverage	1.67	1.83	3.67	4.21	3.62

W ten sam sposób zostały zbadane tablice decyzyjne z jednowartościową decyzją (wyniki w Tabeli 12). Najlepsze trzy heurystyki dla długości to RM (4% różnicy), M (5%) i log (14%). Najlepsze trzy heurystyki dla pokrycia to poly (4% różnicy), log (8%) i maxCov (15%).

Tabela 12: Ogólne średnie relatywne różnice dla heurystyk

		Heuristics				
		poly	log	maxCov	M	RM
Single-valued decisions	Length	0.27	0.14	0.84	0.05	0.04
	Coverage	0.04	0.08	0.15	0.24	0.21
Many-valued decisions	Length	0.29	0.13	0.83	0.02	0.04
	Coverage	0.04	0.08	0.14	0.23	0.20

Z przedstawionych wyników eksperymentalnych można wnioskować, że dla minimalizacji długości należy stosować heurystykę M, można też, heurystykę RM. Dla maksymalizacji pokrycia należy stosować heurystykę poly.

W przyszłych pracach można wykorzystać najlepszą heurystykę dla pokrycia, w algorytmach konstruowania stosunkowo małych systemów reguł pokrywających prawie wszystkie obiekty systemów informacyjnych (zobacz rozdział 7.4).

8 Podsumowanie

Opracowane algorytmy i przedstawione wyniki badań są istotne z punktu widzenia zastosowania reguł jako sposobu reprezentacji wiedzy dla spójnych i niespójnych tablic decyzyjnych (zbiory przybliżone).

Istotnym osiągnięciem jest konstruowanie reguł optymalnych względem różnych kryteriów za pomocą algorytmów dynamicznego programowania. Wiele znanych z literatury algorytmów konstruowania reguł, poza podejściem "brute-force", Boolean reasoning [48] oraz algorytmem Apriori [1], nie zapewnia, że konstruowane reguły są optymalne. Zastosowanie sekwencyjnej optymalizacji reguł względem wybranej kombinacji funkcji kosztu, długość, pokrycie, liczba błędnych zaklasyfikowań, pozwala uzyskać reguły całkowicie optymalne tj. takie, które mają równocześnie, np. minimalną długość, maksymalne pokrycie i minimalną liczbę błędnych zaklasyfikowań. Należy także podkreślić, że zastosowanie sekwencyjnej optymalizacji, podczas konstrukcji regułowych klasyfikatorów, poprawia ich jakość.

Przedstawione rozszerzenia podejścia opartego na dynamicznym programowaniu dla optymalizacji reguł, pomimo dużej złożoności czasowej, mogą zostać zastosowane do analizy mniejszych zbiorów danych, ze względu na ich znaczenie dla celów naukowych i badawczych. Algorytmy te, stanowią dobry punkt odniesienia dla badań nad jakością heurystyk zachłanych.

W ramach poszukiwań algorytmów o stosunkowo niskiej złożoności czasowej, badane były heurystyki zachłanne dla konstruowania krótkich reguł o dużym pokryciu. W tym zakresie uzyskano dobre wyniki, gdyż średnia relatywna różnica odnośnie długości lub pokrycia reguł konstruowanych przez najlepsze heurystyki oraz algorytmy dynamicznego programowania, wynosi 4%.

Na uwagę zasługują także wyniki teoretyczne, m.in. dotyczące algorytmów zachłanych z wagami oraz bez wag. Pokazano, na podstawie badań dotyczących problemu pokrycia zbioru, że algorytmy zachłanne, przy pewnych założeniach o klasie NP, dają wyniki bliskie do najlepszego wielomianowego algorytmu aproksymacyjnego minimalizującego długość częściowych reguł asocjacyjnych

oraz najlepszego wielomianowego algorytmu aproksymacyjnego minimalizującego całkowitą wagę atrybutów zawartych w częściowych regułach asocjacyjnych. Analiza heurystyk zachłanych doprowadziła do nietrywialnych dolnych i górnych oszacowań dotyczących minimalizacji długości częściowych reguł asocjacyjnych oraz minimalizacji całkowitej wagi atrybutów zawartych w regułach.

Istniejące metody konstruowania i optymalizacji reguł zostały poszerzone o podejście oparte na dynamicznym programowaniu oraz o ważne wyniki badań dotyczące algorytmów zachłanych. Wiele wyników teoretycznych zostało potwierdzonych eksperymentalnie. Wymienione w rozdziale czwartym zastosowania zaproponowanych algorytmów, potwierdzają praktyczny aspekt moich badań oraz ich wkład w rozwój teorii zbiorów przybliżonych, zwłaszcza z punktu widzenia reprezentacji wiedzy.

9 Omówienie pozostałych osiągnięć naukowo-badawczych

Moje pozostałe osiągnięcia naukowo-badawcze zostały wyszczególnione w załączniku nr 3: "Wykaz opublikowanych prac naukowych lub twórczych prac zawodowych oraz informacja o osiągnięciach dydaktycznych, współpracy naukowej i popularyzacji nauki".

Poniżej zostaną wymienione wybrane prace, które obejmują realizowane przez mnie kierunki badań podejmowane po doktoracie. Badania te dotyczą m.in. zastosowania reguł, drzew i testów (super-reduktów) jako sposobu reprezentacji wiedzy.

Prace [7, 72] dotyczą konstruowania i optymalizacji reguł decyzyjnych za pomocą algorytmów dynamicznego programowania. W celu zdefiniowania pojęcia reguły przybliżonej, zostały wykorzystane inne miary niepewności dla tablicy decyzyjnej niż liczba nieuporządkowanych par wierszy o różnych decyzjach. Przedstawiono m.in., wyniki dotyczące sekwencyjnej optymalizacji reguł oraz pokazano, że relatywna różnica dotycząca pokrycia przybliżonych reguł decyzyjnych konstruowanych przez algorytm dynamicznego programowania i algorytm zachłanny jest większa niż relatywna różnica dotycząca długości konstruowanych reguł. W pracach [66, 69] zaproponowano, odpowiednio, dla miar niepewności przedstawionych w [7] i [72], zamodyfikowany algorytm dynamicznego programowania. Wyniki eksperymentalne pokazały, że uzyskane wartości pokrycia reguł konstruowanych przez proponowany algorytm są zazwyczaj bliskie wartościom pokrycia reguł optymalnych, a rozmiar grafu jest mniejszy niż rozmiar grafu konstruowanego przez algorytm dynamicznego programowania.

W artykule [6] przedstawiono podejście oparte na dynamicznym programowaniu dla sekwencyjnej optymalizacji reguł niedeterministycznych (wzbraniających). Eksperymenty potwierdziły istnienie całkowicie optymalnych reguł wzbraniających dla tablic decyzyjnych, tj. reguł o minimalnej długości i maksymalnym pokryciu.

Praca [8] przedstawia porównanie klasyfikatorów opartych na dokładnych i przybliżonych regułach optymalizowanych za pomocą algorytmów dynamicznego programowania. Pokazano także, że zastosowanie dwukryterialnej optyma-

lizacji (pokrycie + długość lub długość + pokrycie) reguł przybliżonych poprawia jakość pracy klasyfikatorów, w porównaniu z jednokryterialną optymalizacją reguł.

Prace [68, 71], są kontynuacją tematyki przedstawionej w [67] i dotyczą zastosowania podejścia opartego na dynamicznym programowaniu dla optymalizacji reguł asocjacyjnych względem pokrycia i długości, także dla niespójnych tablic decyzyjnych.

Praca [73] przedstawia porównanie klasyfikatorów opartych na deterministycznych i niedeterministycznych regułach decyzyjnych.

W pracach [45, 46] przedstawiono zastosowanie reguł konstruowanych przez algorytm zachłanny w procesach wnioskowania w bazach wiedzy.

Jestem także współautorem prac dotyczących badania algorytmu zachłannego dla konstruowania testów (super-reduktów) [12] oraz drzew decyzyjnych [13] dla tablic decyzyjnych z wielowartościową decyzją. Przedstawione wyniki dotyczą m.in., odpowiednio, dokładności algorytmu odnośnie głębokości konstruowanych drzew decyzyjnych oraz liczebności konstruowanych testów.

Obecnie, przyjęty został do druku (Transaction on Rough Sets) artykuł [14] dotyczący konstruowania przybliżonych reguł decyzyjnych za pomocą algorytmu zachłannego dla tablic decyzyjnych z wielowartościową decyzją. Celem pracy jest pokazanie, że proponowane podejście dla konstruowania reguł jest użyteczne z punktu widzenia reprezentacji wiedzy. Praca ta stanowi istotne rozszerzenie publikacji [21] oraz wyników badań umieszczonych w [41]. Poza wynikami teoretycznymi praca zawiera wiele wyników eksperymentalnych, m.in. dotyczących porównania liczby różnych reguł oraz długości reguł dla proponowanego podejścia oraz podejścia opartego na uogólnionej decyzji, znanego w teorii zbiorów przybliżonych.

Literatura

1. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In: SIGMOD '93. ACM (1993) 207–216
2. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In Bocca, J.B., Jarke, M., Zaniolo, C., eds.: VLDB '94. Morgan Kaufmann (1994) 487–499
3. Alsolami, F., Amin, T., Chikalov, I., Moshkov, M.: Multi-stage optimization of decision rules for knowledge discovery and representation. Knowledge and Information Systems (2015) (submitted)
4. Alsolami, F., Amin, T., Chikalov, I., Moshkov, M., Zielosko, B.: Dynamic programming approach for construction of association rule systems. In Suraj, Z., Czaja, L., eds.: Proceedings of the 24th International Workshop on Concurrency, Specification and Programming 2015. CEUR-WS.org (2015) 12–21
5. Alsolami, F., Amin, T., Moshkov, M., Zielosko, B.: Comparison of heuristics for optimization of association rules. In Suraj, Z., Czaja, L., eds.: Proceedings of the 24th International Workshop on Concurrency, Specification and Programming 2015. CEUR-WS.org (2015) 4–11
6. Alsolami, F., Chikalov, I., Moshkov, M., Zielosko, B.: Length and coverage of inhibitory decision rules. In Nguyen, N.T., Hoang, K., Jędrzejowicz, P., eds.: ICCCI 2012. Volume 7654 of LNCS. Springer (2012) 325–334

7. Amin, T., Chikalov, I., Moshkov, M., Zielosko, B.: Dynamic programming approach for partial decision rule optimization. *Fundam. Inform.* **119**(3-4) (2012) 233–248
8. Amin, T., Chikalov, I., Moshkov, M., Zielosko, B.: Classifiers based on optimal decision rules. *Fundam. Inform.* **127**(1-4) (2013) 151–160
9. Amin, T., Chikalov, I., Moshkov, M., Zielosko, B.: Dynamic programming approach to optimization of approximate decision rules. *Inf. Sci.* **221** (2013) 403–418
10. Amin, T., Chikalov, I., Moshkov, M., Zielosko, B.: Relationships between length and coverage of decision rules. *Fundam. Inform.* **129**(1-2) (2014) 1–13
11. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository. <http://www.ics.uci.edu/~mllearn/> (2007)
12. Azad, M., Chikalov, I., Moshkov, M., Zielosko, B.: Greedy algorithms for construction of approximate tests for decision tables with many-valued decisions. *Fundam. Inform.* **120**(3-4) (2012) 231–242
13. Azad, M., Chikalov, I., Moshkov, M., Zielosko, B.: A greedy algorithm for construction of decision trees for tables with many-valued decisions - A comparative study. *Fundam. Inform.* **128**(1-2) (2013) 1–15
14. Azad, M., Moshkov, M., Zielosko, B.: Greedy algorithm for the construction of approximate decision rules for decision tables with many-valued decisions. In: *Transactions on Rough Sets. LNCS*. Springer (2016) (accepted).
15. Błaszczyński, J., Słowiński, R., Szelaż, M.: Sequential covering rule induction algorithm for variable consistency rough set approaches. *Inf. Sci.* **181**(5) (2011) 987–1002
16. Bonates, T., Hammer, P.L., Kogan, A.: Maximum patterns in datasets. *Discrete Applied Mathematics* **156**(6) (2008) 846–861
17. Borgelt, C.: Simple algorithms for frequent item set mining. In Koronacki, J., Raś, Z.W., Wierchoń, S.T., Kacprzyk, J., eds.: *Advances in Machine Learning II*. Volume 263 of *Studies in Computational Intelligence*. Springer (2010) 351–369
18. Borgelt, C.: Frequent item set mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **2**(6) (2012) 437–456
19. Boutell, M.R., Luo, J., Shen, X., Brown, C.M.: Learning multi-label scene classification. *Pattern Recognition* **37**(9) (2004) 1757–1771
20. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: *Classification and Regression Trees*. Chapman & Hall, New York (1984)
21. Chikalov, I., Zielosko, B.: Decision rules for decision tables with many-valued decisions. In Yao, J., Ramanna, S., Wang, G., Suraj, Z., eds.: *RSKT 2011*. Volume 6954 of *LNCS*. Springer (2011) 763–768
22. Clark, P., Niblett, T.: The CN2 induction algorithm. *Mach. Learn.* **3**(4) (1989) 261–283
23. Delimata, P., Moshkov, M.J., Skowron, A., Suraj, Z.: Inhibitory Rules in Data Analysis: A Rough Set Approach. Volume 163 of *Studies in Computational Intelligence*. Springer (2009)
24. Feige, U.: A threshold of $\ln n$ for approximating set cover (preliminary version). In Miller, G.L., ed.: *28th Annual ACM Symposium on the Theory of Computing*. ACM (1996) 314–318
25. Herawan, T., Deris, M.M.: A soft set approach for association rules mining. *Knowledge-Based Systems* **24**(1) (2011) 186–195
26. Moshkov, M.: *Decision Trees. Theory and Applications*. Nizhny Novgorod University Publishers, Nizhny Novgorod (1994) (in Russian).
27. Moshkov, M.: On global Shannon functions of two-valued information systems. In: *Proc. Fourth Int'l Workshop on Rough Sets, Fuzzy Sets and Machine Discovery*, Tokyo, Japan. (1996) 142–143

28. Moshkov, M.: Unimprovable upper bounds on complexity of decision trees over information systems. *Foundations of Computing and Decision Sciences* **21** (1996) 219–231
29. Moshkov, M.: Classification of infinite information systems depending on complexity of decision trees and decision rule systems. *Fundam. Inform.* **54**(4) (2003) 345–368
30. Moshkov, M.: Greedy algorithm for decision tree construction in context of knowledge discovery problems. In Tsumoto, S., Słowiński, R., Komorowski, H.J., Grzymała-Busse, J.W., eds.: *RSTC 2004*. Volume 3066 of LNCS. Springer (2004) 192–197
31. Moshkov, M.: On the class of restricted linear information systems. *Discrete Mathematics* **307**(22) (2007) 2837–2844
32. Moshkov, M., Chikalov, I.: On algorithm for constructing of decision trees with minimal depth. *Fundam. Inform.* **41**(3) (2000) 295–299
33. Moshkov, M., Piliszczuk, M., Zielosko, B.: On construction of partial reducts and irreducible partial decision rules. *Fundam. Inform.* **75**(1-4) (2007) 357–374
34. Moshkov, M., Piliszczuk, M., Zielosko, B.: On partial covers, reducts and decision rules with weights. In Peters, J.F., Skowron, A., Düntsch, I., Grzymała-Busse, J.W., Orłowska, E., Polkowski, L., eds.: *Transactions on Rough Sets VI*. Volume 4374 of LNCS. Springer (2007) 211–246
35. Moshkov, M., Piliszczuk, M., Zielosko, B.: On partial covers, reducts and decision rules. In Peters, J.F., Skowron, A., eds.: *Transactions on Rough Sets VIII*. Volume 5084 of LNCS. Springer (2008) 251–288
36. Moshkov, M., Piliszczuk, M., Zielosko, B.: *Partial Covers, Reducts and Decision Rules in Rough Sets. Theory and Applications*. Volume 145 of *Studies in Computational Intelligence*. Springer (2008)
37. Moshkov, M., Piliszczuk, M., Zielosko, B.: Universal problem of attribute reduction. In Peters, J.F., Skowron, A., Rybiński, H., eds.: *Transactions on Rough Sets IX*. Volume 5390 of LNCS. Springer (2008) 187–199
38. Moshkov, M., Piliszczuk, M., Zielosko, B.: Greedy algorithm for construction of partial association rules. *Fundam. Inform.* **92**(3) (2009) 259–277
39. Moshkov, M., Piliszczuk, M., Zielosko, B.: Greedy algorithms with weights for construction of partial association rules. *Fundam. Inform.* **94**(1) (2009) 101–120
40. Moshkov, M., Zielosko, B.: Decision rule systems and decision trees over finite binary information systems (plenary talk). In: *The Sixth Rough Set Theory Workshop*, University of Warsaw, The Faculty of Mathematics, Informatics and Mechanics, Poland, Warsaw, June 29, 2015
41. Moshkov, M., Zielosko, B.: *Combinatorial Machine Learning. A Rough Set Approach*. Volume 360 of *Studies in Computational Intelligence*. Springer (2011)
42. Nguyen, H.S.: Approximate boolean reasoning: foundations and applications in data mining. In Peters, J.F., Skowron, A., eds.: *Transactions on Rough Sets V*. Volume 4100 of LNCS. Springer (2006) 334–506
43. Nguyen, H.S., Nguyen, S.H.: Rough sets and association rule generation. *Fundam. Inform.* **40**(4) (1999) 383–405
44. Nguyen, H.S., Ślęzak, D.: Approximate reducts and association rules - correspondence and complexity results. In Zhong, N., Skowron, A., Ohsuga, S., eds.: *RSFD-GrC 1999*. Volume 1711 of LNCS. Springer (1999) 137–145
45. Nowak, A., Zielosko, B.: Clustering of partial decision rules. In Cyran, K.A., Kozielski, S., Peters, J.F., Stańczyk, U., Wakulicz-Deja, A., eds.: *ICMMI 2009*. Volume 59 of *Advances in Intelligent and Soft Computing*. (2009) 183–190

46. Nowak, A., Zielosko, B.: Inference processes on clustered partial decision rules. In Kłopotek, M.A., Przepiórkowski, A., Wierzchoń, S.T., eds.: *Recent Advances in Intelligent Information Systems*. Academic Publishing House EXIT (2009) 579–588
47. Pawlak, Z.: *Rough Sets—Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers, Dordrecht (1991)
48. Pawlak, Z., Skowron, A.: Rudiments of rough sets. *Inf. Sci.* **177**(1) (2007) 3–27
49. Quinlan, J.R.: *C4.5: Programs for Machine Learning*. Morgan Kaufmann (1993)
50. Savasere, A., Omieciński, E., Navathe, S.B.: An efficient algorithm for mining association rules in large databases. In Dayal, U., Gray, P.M.D., Nishio, S., eds.: *VLDB 1995*. Morgan Kaufmann (1995) 432–444
51. Skowron, A., Rauszer, C.: The discernibility matrices and functions in information systems. In Słowiński, R., ed.: *Intelligent Decision Support. Handbook of Applications and Advances of the Rough Set Theory*. Kluwer Academic Publishers (1992) 331–362
52. Skowron, A.: Rough sets in KDD - plenary talk. In Z. Shi, B. Faltings, M. Musen, eds.: *Proc. 16th IFIP. World Computer Congress*. Publishing House of Electronic Industry (2000) 1–14
53. Slavík, P.: A tight analysis of the greedy algorithm for set cover. In: *Proc. of the 28th Annual ACM Symposium on the Theory of Computing*. ACM (1996) 435–441
54. Slavík, P.: *Approximation algorithms for set cover and related problems*. Ph.D. Thesis. University of New York at Buffalo (1998)
55. Ślęzak, D.: Normalized decision functions and measures for inconsistent decision tables analysis. *Fundam. Inform.* **44**(3) (2000) 291–319
56. Ślęzak, D.: Approximate entropy reducts. *Fundam. Inform.* **53**(3-4) (2002) 365–390
57. Słowiński, R., Greco, S.: Measuring attractiveness of rules from the viewpoint of knowledge representation, prediction and efficiency of intervention. In Szczepaniak, P.S., Kacprzyk, J., Niewiadomski, A., eds.: *Advances in Web Intelligence*. Volume 3528 of LNCS. Springer (2005) 11–22
58. Srikant, R., Agrawal, R.: Mining quantitative association rules in large relational tables. In: *ACM SIGMOD'96*, ACM (1996) 1–12
59. Stefanowski, J., Vanderpooten, D.: Induction of decision rules in classification and discovery-oriented perspectives. *Int. J. Intell. Syst.* **16**(1) (2001) 13–27
60. Wiczorek, A., Słowiński, R.: Generating a set of association and decision rules with statistically representative support and anti-support. *Inf. Sci.* **277** (2014) 56–70
61. Wróblewski, J.: Ensembles of classifiers based on approximate reducts. *Fundam. Inform.* **47**(3-4) (2001) 351–360
62. Zhou, Z.H., Zhang, M.L., Huang, S.J., Li, Y.F.: Multi-instance multi-label learning. *Artif. Intell.* **176**(1) (2012) 2291–2320
63. Zielosko, B.: On partial decision rules. In Czaja, L., ed.: *Proceedings of the Concurrency, Specification and Programming, Ruciane-Nida, Poland, September 28-30, 2005*. Warsaw University (2005) 598–609
64. Zielosko, B.: Greedy algorithm for construction of partial association rules. *Studia Informatica* (31) (2010) 225–236 (in Polish).
65. Zielosko, B.: Optimization of approximate decision rules relative to coverage. In Kozielski, S., Mrózek, D., Kasprowski, P., Małysiak-Mrózek, B., Kostrzewa, D., eds.: *BDAS 2014*. Volume 424 of CCIS. Springer (2014) 170–179
66. Zielosko, B.: Optimization of decision rules relative to coverage - comparative study. In Kryszkiewicz, M., Cornelis, C., Ciucci, D., Medina-Moreno, J., Motoda, H., Raś, Z.W., eds.: *RSEISP 2014*. Volume 8537 of LNCS., Springer (2014) 237–247

67. Zielosko, B.: Global optimization of exact association rules relative to coverage. In Kryszkiewicz, M., Bandyopadhyay, S., Rybiński, H., Pal, S.K., eds.: PReMI 2015. Volume 9124 of LNCS. Springer (2015) 428–437
68. Zielosko, B.: Global optimization of exact association rules relative to length. In Suraj, Z., Czaja, L., eds.: Proceedings of the 24th International Workshop on Concurrency, Specification and Programming 2015. CEUR-WS.org (2015) 237–247
69. Zielosko, B.: Optimization of decision rules relative to coverage - comparison of greedy and modified dynamic programming approaches. In Gruca, A., Brachman, A., Kozielski, S., Czachórski, T., eds.: ICMMI 2015. Volume 391 of Advances in Intelligent Systems and Computing., Springer (2015) 639–650
70. Zielosko, B.: Sequential optimization of γ -decision rules relative to length, coverage and number of misclassificationss. In Peters, J.F., Skowron, A., Ślęzak, D., Nguyen, H.S., Bazan, J.G., eds.: Transactions on Rough Sets XIX. Volume 8988 of LNCS. Springer (2015) 65–82
71. Zielosko, B.: Application of dynamic programming approach to optimization of association rules relative to coverage and length. Fundam. Inform. (2016) (submitted).
72. Zielosko, B., Chikalov, I., Moshkov, M., Amin, T.: Optimization of decision rules based on dynamic programming approach. In Faucher, C., Jain, L.C., eds.: Innovations in Intelligent Machines (4). Volume 514 of Studies in Computational Intelligence. Springer (2014) 369–392
73. Zielosko, B., Marszał-Paszek, B., Paszek, P.: Partial and nondeterministic decision rules in classification process. In Kłopotek, M.A., Przepiórkowski, A., Wierchoń, S.T., eds.: Recent Advances in Intelligent Information Systems. Academic Publishing House EXIT (2009) 629–638
74. Zielosko, B., Piliszczuk, M.: Greedy algorithm for attribute reduction. Fundam. Inform. 85(1-4) (2008) 549–561

Beata Zielosko