Politechnika Wrocławska
Wydział Informatyki i Zarządzania
Instytut Informatyki

Rozprawa doktorska

PRZYROSTOWA EKSTRAKCJA WIEDZY Z DANYCH DLA OBIEKTÓW NIESTACJONARNYCH

Jakub Mikołaj Tomczak

Promotor: prof. dr hab. inż. Jerzy Świątek

Wrocław 2012

Podziękowania

Na wstępie chciałbym podziękować mojemu promotorowi, prof. dr hab. inż. Jerzemu Świątkowi, za wszelką pomoc i opiekę naukową, którą sprawuje nade mną od momentu pisania pracy magisterskiej po dzień dzisiejszy. Jestem również niezmiernie wdzięczny prof. dr hab. inż. Adamowi Grzechowi za cenne uwagi oraz cierpliwość podczas wielu seminariów i spotkań.

Ponadto chciałbym złożyć podziękowania moim kolegom z Instytutu Informatyki, przede wszystkim Adamowi Gonczarkowi za udzieloną pomoc oraz liczne dyskusje, które przenosiły się również poza mury Politechniki, ale także drowi inż. Krzysztofowi Brzostowskiemu za wskazanie zastosowania w dziedzinie diabetologii, Piotrowi Rygielskiemu za implementację środowiska symulacyjnego systemu zorientowanego na usługi, Maciejowi Ziębie i Maciejowi Drwalowi za wspólne poznawanie świata nauki, Pawłowi Stelmachowi za konstruktywne sprzeczki odnośnie systemów o paradygmacie SOA, oraz drowi inż. Jarosławowi Drapale i drowi inż. Pawłowi Świątkowi za niepowtarzalny nastrój pracy.

Last but not least wielkie słowa uznania należą się moim Rodzicom oraz Bratu, ponieważ od początku wspierali mnie we wszystkich działaniach i pocieszali w trudnych chwilach.

Pracę chciałbym zadedykować mojemu wujkowi, Ś.P. Zdzisławowi Bubnickiemu, ponieważ bez niego nigdy nie znalazłbym się we Wrocławiu i najprawdopodobniej nie wybrał obecnej drogi życiowej.

Część niniejszej pracy jest współfinansowana ze środków Unii Europejskiej poprzez Europejski Fundusz Rozwoju Regionalnego w ramach Programu Operacyjnego Innowacyjna Gospodarka na lata 2007-2013, numer projektu: POIG.01.03.01-00-008/08.

Część niniejszej pracy jest wykonana w ramach stypendium współfinansowanego przez Unię Europejską w ramach Europejskiego Funduszu Społecznego.

Spis treści

Rozdział 1

Wstęp

1.1 Wprowadzenie

Obecnie w większości systemów informatycznych zbierane są coraz większe wolumeny danych, które napływają w strumieniu danych (ang. *datastream*) [44, 47, 49, 114] oraz są przesyłane [56, 133], agregowane oraz przetwarzane [24]. Przetwarzanie danych wiąże się z ekstrakcją wiedzy, czyli odkrywaniem zależności opisujących obserwowany obiekt. Istnieje wiele zastosowań, w których wiedza odgrywa kluczową rolę w procesie podejmowania decyzji, np. w systemach ekspertowych diagnostyki medycznej [19, 90, 145, 148], systemach zorientowanych na usługi [20, 43, 130, 150, 146], systemach automatycznej analizy zachowania klientów systemów informatycznych [14, 30, 82, 160], systemach produkcyjnych [87], systemach zarządzania i analiz finansowych [24, 44, 87], analizie sieci społecznych [167], zarządzanie ruchem teleinformatycznym [85], systemach sterowania [28], informatycznych systemach edukacji [140].

Jednak ze względu na stopień złożoności problemów przetwarzania danych w celu otrzymania wiedzy, eksperci dziedzinowi nie są w stanie podać rozwiązań w zadowalającym czasie oraz z zadowalającą jakością. W literaturze zjawisko to określane jest mianem wąskiego gardła procesu pozyskiwania wiedzy (ang. *knowledge acquisition bottleneck*) [91, 104]. Dlatego też rośnie zapotrzebowanie na automatyczne pozyskiwanie wiedzy dla wspomagania procesów decyzyjnych.

Kolejnym wyzwaniem w procesach podejmowania decyzji jest zmienność własności

obiektu (zjawiska, procesu) [152], np. zmienny stan pacjenta, zmienny strumień żądań do systemu usługowego. Aby móc podejmować decyzje w oparciu o aktualny stan wiedzy o obiekcie, należy zaproponować metody, które pozwalają na

- szybkie przetwarzanie strumieni danych;
- otrzymywanie zwięzłego opisu obiektu;
- walidację i uaktualnianie wiedzy na podstawie nowo pojawiających się obserwacji.

Dziedziną informatyki, która zajmuje się opracowaniem algorytmów ekstrakcji wiedzy o obiekcie jest uczenie maszynowe (ang. *machine learning*) [13, 34, 51, 106].

1.2 Opis problemu

1.2.1 Wiedza

Poprzez wiedzę rozumie się zwięzły opis dotyczący obiektu, wyrażony w wybranej reprezentacji, który został sformułowany na podstawie obserwacji.

Do podanej definicji czasami dodaje się, że wiedza może mieć charakter deklaratywny, czyli wyraża fakty, oraz proceduralny, czyli przedstawia procedury [34, 68]. Opis dotyczący obiektu może również uwzględniać stopień pewności wiedzy [27]. Niektórzy autorzy wskazują też, że wiedza musi być zrozumiała dla maszyny i człowieka [34, 79].

W literaturze przedmiotu podkreśla się, iż celem wiedzy jest uogólnienie obserwacji oraz uwzględnienie istotnych, z punktu widzenia procesu podejmowania decyzji, informacji tak, aby ich złożoność opisu była mniejsza od złożoności opisu samych obserwacji [47]. Ciąg obserwacji określany będzie jako *ciąg uczący*.

W dziedzinie informatyki często stosuje się zamiennie słowo „model” i „wiedza” [24], mimo że pojęcia te mają nieco odmienne znaczenia. W niniejszej pracy przyjmuje się, że model jest pojęciem szerszym i abstrakcyjnym, natomiast wiedza dotyczy konkretnych zależności opisujących obiekt. Tym niemniej oba określenia mogą być używane zamiennie.

W dziedzinie badań systemowych [21, 22, 143], jak również uczenia maszynowego [13], wyszczególnia się

- modele parametryczne – model opisujący obiekt znany jest z dokładnością do ustalonej liczby parametrów;
- modele nieparametryczne – model opisujący obiekt jest wyrażony za pomocą parametrów, których liczba zależy od liczby obserwacji.

Zarówno dla modeli parametrycznych i nieparametrycznych wiedza zawarta jest w konkretnych wartościach parametrów.

Obserwacje obiektu mogą dotyczyć różnego rodzaju informacji [51, 139], tj.

- informacji nominalnych (symboliczne) – informacje przyjmują wartości z dyskretnego zbioru wartości, na którym nie ma nałożonego porządku (np. grupa krwi A+, AB-, 0-, itd.);
- informacji porządkowych – informacje przyjmują wartości z dyskretnego zbioru wartości, w którym można wprowadzić porządek (np. mały, średni, duży);
- informacji strukturalnych – informacje reprezentują relacje dotyczące obiektu np. za pomocą struktury drzewa;
- informacji mierzalnych (numeryczne) – informacje przyjmują wartości rzeczywiste.

W dalszym ciągu pracy skupiamy się na informacjach nominalnych i porządkowych.

1.2.2 Reprezentacje wiedzy

W celu wyrażania wiedzy stosuje się różnego rodzaju *klasy modeli* (zwane też *reprezentacjami*). Wyszczególnia się następujące klasy modeli:

- parametryczne:
  - wyrażenia funkcyjne (np. [21, 29]) – klasa modeli odpowiada przestrzeni funkcji o zadanej postaci, np. sieci neuronalne [129];
  - grafy i sieci (np. [29]) – obiekty matematyczne, które posiadają zbiór wierzchołków oraz krawędzi (łuków);

  - modele rozmyte i niepewne (np. [27, 129]) – zależność opisująca obiekt jest wyrażona za pomocą opisów niepewnych, czyli funkcji przynależności lub rozkładu niepewności;
  - parametryczne modele probabilistyczne (np. [13, 70]) – wiedza reprezentowana jest za pomocą modelu generującego (ang. *generative models*), który może być przedstawiony jako np. sieć Bayesa czy markowskie pole losowe, lub poprzez model dyskryminacyjny (ang. *discriminative models*), np. regresja logistyczna, modele typu ensemble;

- nieparametryczne (strukturalne):
  - reguły (wyrażenia logiczne) (np. [22, 23, 25, 29, 35, 34, 51, 68, 106, 164]) – regułami nazywamy wyrażenia, które dotyczą zazwyczaj informacji nominalnych i porządkowych, i są reprezentowane w koniunkcyjnej postaci normalnej (ang. *conjuctive normal form*, CNF) lub dysjunkcyjnej postaci normalnej (ang. *disjunctive normal form*, DNF) w logice z atrybutami [93, 102];
  - drzewa decyzyjne (np. [29, 34, 35, 51, 83, 106, 164])– reprezentacja zbliżona do reguł, jednak wiedza przedstawiana jest za pomocą grafu w postaci drzewa, gdzie decyzje znajdują się w liściach, zaś w pozostałych węzłach określane są warunki dot. zmiennych wejściowych;
  - wyrażenia logiczne wyższego rzędu (np. [91, 106]) – modele wyrażane w języku predykatów pierwszego lub wyższego rzędu;
  - gramatyki formalne (np. [29]) – reguły lub automaty skończone określające język formalny;
  - schematy i ramy (np. [68]) – obiekty, które są złożeniem faktów oraz procedur postępowania (akcji);
  - sieci semantyczne i ontologie (np. [29, 35, 68]) – reprezentacje grafowe, które uwzględniają informacje semantyczne oraz relacje między pojęciami;
  - zbiory przybliżone (np. [116, 129]) – wiedza otrzymywana jest przy pomocy dolnej i górnej aproksymacji wybranego pojęcia w oparciu o teorię mnogości.

- maszyny wektorów wspierających (ang. *support vector machines*) [156] – klasyfikacja lub regresja wyznaczana jest na podstawie wybranych obserwacji, które minimalizują wartość zadanego kryterium optymalizacji;
- nieparametryczne modele probabilistyczne – modele probabilistyczne, które definiują rozkład prawdopodobieństwa na przestrzeń funkcji, np. *procesy Gaussa* [126], *procesy Dirichleta* [71].

Zazwyczaj modele nieparametryczne są wykorzystywane do uwzględniania informacji porządkowych i nominalnych, oraz wyrażane są za pomocą reprezentacji symbolicznych (np. operatory logiczne), dlatego też w wielu dziedzinach ten rodzaj reprezentacji wiedzy uznawany jest za faktyczną wiedzę. Jednak w rozumieniu wiedzy jako konkretnego modelu, tj. konkretnych wartości parametrów lub konkretnej struktury, wszystkie reprezentacje podane powyżej mogą służyć do reprezentowania wiedzy.

W niniejszej pracy skupiamy się wyłącznie na reprezentacji regułowej, tzn. na wyrażeniach logicznych w logice z atrybutami, które uwzględniają informacje nominalne i porządkowe, bez informacji numerycznych i strukturalnych. W literaturze wyszczególnia się *reguły asocjacyjne* (ang. *association rules*) [35, 164], które określają związki między informacjami, oraz *reguły decyzyjne* (ang. *decision rules*), zwane też *klasyfikacyjnymi*, czy *produktowymi* [27, 29, 34, 106, 164], które wiążą wejście obiektu z jego wyjściem. Często zamiast o obiekcie mówi się o *koncepcie* (np. [106]). Wówczas opis konceptu (pojęcia) określają informacje wejściowe, natomiast wyjście oznacza, czy podane informacje dotyczą konceptu, czy też nie.

1.2.3 Proces ekstrakcji wiedzy

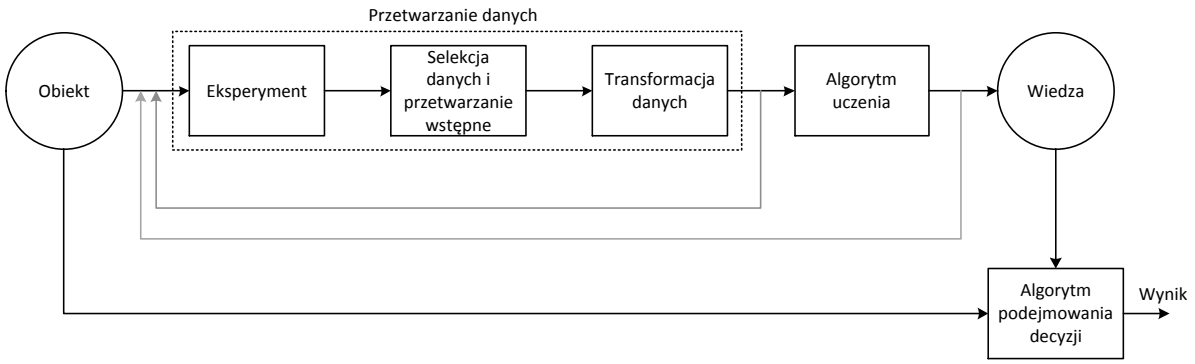
Ogólnie mówiąc proces ekstrakcji wiedzy polega na pobraniu obserwacji o rozpatrywanym obiekcie i przetworzeniu ich w celu uzyskania wiedzy. Proces ekstrakcji wiedzy, którego schemat przedstawiono na rysunku 1.1, składa się z następujących kroków [35, 44, 107]:

1. *Przeprowadzenie eksperymentu* – obserwacje o obiekcie zbierane są poprzez wykonanie eksperymentu (biernego lub czynnego) [21].

2. *Selekcja danych i przetwarzanie wstępne* – z otrzymanych obserwacji wybierane są te, które nadają się do dalszego przetwarzania, oraz dokonywana jest wstępna obróbka danych, np. usuwanie szumu, brakujących wartości, dyskretyzacja informacji, normalizacja informacji [80, 139].

3. *Transformacja danych* – obserwacje zostają przetworzone poprzez ekstrakcję lub selekcję cech.

4. *Algorytm uczenia* – po procesie przetwarzania danych dokonywana zostaje analiza danych i formułowana jest wiedza.



Rysunek 1.1: Proces ekstrakcji wiedzy z zaznaczonymi krokami.

W niniejszej pracy zakładamy, że kroki związane z przetwarzaniem danych (przeprowadzenie eksperymentu, przetwarzanie wstępne i transformacja danych) został pomyślnie wykonane i skupimy się na opracowaniu odpowiednich algorytmów uczenia.

Ekstrakcję wiedzy wyrażonej w reprezentacji regułowej w literaturze przedmiotu określa się jako *indukcję reguł* (ang. *rules induction*) [34, 87].

1.2.4 Techniki uczenia. Uczenie przyrostowe

Istnieją dwa główne paradygmaty wnioskowania [16, 120, 137], tj. dedukcja i redukcja. Główną metodą wnioskowania redukcyjnego jest *indukcja* [103]. Wnioskowanie indukcyj-

ne polega na generalizowaniu obserwacji, w wyniku czego otrzymywany jest opis obiektu (wiedza).

W [120] zarzuca się, że indukcja jest mniej *naukowa* niż dedukcja, która jest jedyną poprawną metodą wnioskowania. Jednakże w literaturze filozofowie nauki nie osiągnęli jednoznacznego stanowiska, które odrzucałoby indukcję jako metodę naukową; co więcej, istnieje wiele głosów ten pogląd podważający [33]. W przełomowej pracy [156], w oparciu o analizę procesów empirycznych, podano statystyczne własności algorytmów indukcyjnego uczenia jako narzędzia pozyskiwania wiedzy, tym samym wskazując na formalne własności dotyczące skuteczności stosowania indukcji.

Indukcyjna ekstrakcja wiedzy z danych, zwana też *uczeniem*, jest zdefiniowana w następujący sposób [91]:
Posiadając wiedzę dziedzinową, ciąg obserwacji, kryterium oraz klasę modeli, znajdź nieznane wartości parametrów, które „najlepiej" odzwierciedlają obiekt (zjawisko, koncept).

Wyrażenie *najlepiej* w powyższej definicji oznacza, że zgodnie z określonym kryterium otrzymana wiedza, tzn. konkretne wartości parametrów lub konkretna struktura, odzwierciedla zbiór danych, czyli pokrywa go w sposób spójny i nie stoi w sprzeczności z wiedzą dziedzinową.

Wyszczególnia się dwie główne techniki uczenia [34]:

1. **Uczenie wsadowe** (ang. *batch learning*) – ciąg uczący przetwarzany jest w całości;
2. **Uczenie przyrostowe** (na bieżąco) (ang. *incremental learning*) [11, 75] – obserwacje są przetwarzane przez algorytm uczenia sekwencyjnie.

Dodatkowo, gdy celem uczenia jest nadążanie za charakterystyką obiektu zależną od czasu, w uczeniu przyrostowym należy zaproponować odpowiedni mechanizm *zapominania* [98, 96, 113, 132]:

- zapominanie czasowe (ang. *explicit forgetting*) – polega na zapominaniu najstarszych obserwacji i wyszczególnia się zapominanie:
  - z oknem przesuwnym o stałej długości (ang. *forgetting with constant shifting window*) – wiedza jest uaktualniana na podstawie ostatnich danych zawartych

w tzw. oknie, natomiast nowa obserwacja powoduje usunięcie najstarszej obserwacji z okna;

- z oknem przesuwnym o zmiennej długości (ang. *forgetting with changing shifting window*) – wiedza uaktualniana jest na podstawie ostatnich obserwacji zawartych w tzw. oknie, jednak długość okna określana jest za pomocą dodatkowej metody;
- wykładnicze (ang. *exponential forgetting*) – im starsza obserwacja, tym jej wkład do uaktualniania modelu jest mniejszy.

- zapominanie wybiórcze (ang. *implicit forgetting*) – polega na zapominaniu wybranych obserwacji lub części składowych wiedzy, niekoniecznie najstarszych.

Mechanizm zapominania wybiórczego wymusza stosowanie innych technik, np. usuwanie obserwacji, które stoją w sprzeczności z najnowszą obserwacją (np. jak w AQ-PM [98]), zapominanie lokalne, które opiera się na rozkładach prawdopodobieństwa [131, 132].

Połączenie uczenia przyrostowego z mechanizmem zapominania prowadzi do ekstrakcji wiedzy z adaptacją [38, 113].

1.2.5 Niestacjonarność

W praktyce często spotykane są obiekty, których właściwości zmieniają się w czasie ich trwania z powodu oddziaływań ze środowiskiem, np. stan zdrowia pacjenta, który zależy od sposobu leczenia, diety, aktywności fizycznej, trybu życia. Środowisko, które zakłada się, że jest nieobserwowalne, nazywane jest kontekstem (ang. *hidden context*) [60, 81, 100, 161].

Pojawia się zatem konieczność uwzględnienia w ekstrakcji wiedzy wpływu kontekstu na obiekt. Obiekty o właściwościach zależnych od zmiennego kontekstu nazywa się *obiektami niestacjonarnymi*. Zmienność obiektu w czasie może przebiegać w dwojaki sposób [152, 163]:

1. Zmiana stopniowa (ang. *gradual change*) – obiekt ze względu na kontekst zmienia się w sposób ciągły, np. zużywanie się elementów układu elektronicznego.

2. Zmiana nagła (ang. *abrupt change*) – w przedziale poprzedzającym zmianę oraz po dokonaniu się zmiany kontekstu właściwości obiektu są stałe, np. moc obliczeniowa systemu komputerowego po wymianie sprzętu.

W celu rozwiązania problemu ekstrakcji wiedzy o obiekcie niestacjonarnym stosuje się dwa podejścia [21]:

1. Podejście z modelem niestacjonarnym.

2. Podejście z modelem stacjonarnym.

W pierwszym przypadku zakłada się, że model odzwierciedla zachowanie zmiennej charakterystyki obiektu, czyli rozpatruje się model zależny od czasu. Przykładowo, dla starzejących się elementów układu elektronicznego, można przyjąć model o parametrach zmiennych w czasie. Wówczas, dla zadanego ciągu uczącego, wybiera się najlepszy model ze względu na wybrane kryterium. Główną wadą takiego podejścia jest złożoność wyznaczenia modelu oraz trudność w wykorzystaniu takiego modelu w procesie podejmowania decyzji [21].

W przypadku, gdy zmiana kontekstu ma charakter nagły, wygodnym rozwiązaniem jest stosowanie modelu stacjonarnego, tzn. klasa modeli nie uwzględnia zależności od czasu. Wówczas w procesie uczenia model jest uaktualniany z wykorzystaniem nowo pojawiających się obserwacji. Model stacjonarny jest na ogół prościej wyznaczyć w wyniku ekstrakcji wiedzy w porównaniu z modelem niestacjonarnym (tzn. prostsze są algorytmy uczenia), łatwiejsze może być również zaproponowanie algorytmu podejmowania decyzji dla takiego modelu [21].

1.2.6 Ogólne sformułowanie problemu ekstrakcji wiedzy

W obiekcie wyszczególnia się:

1. Wejście (zwane *atrybutami* lub *cechami*) $\mathbf{u} = [u^1 \ u^2 \ \ldots u^D]^{\mathrm{T}} \in \mathscr{U}$, gdzie $\mathscr{U} = \mathscr{U}_1 \times \mathscr{U}_2 \times \ldots \times \mathscr{U}_D$, oraz, dla każdego $d = 1, 2, \ldots, D$, $\mathrm{card}\{\mathscr{U}_d\} = K_d < \infty$ i oznaczmy $\sum_{d=1}^{D} K_d = K$.

2. Wyjście (zwane *klasą*) $y \in \mathscr{Y}$, gdzie $\mathscr{Y} = \{0, 1, \ldots, (Y-1)\}$, $\mathrm{card}\{\mathscr{Y}\} = Y$.

Na obiekt działa zmienny w czasie kontekst, $c_m \in \mathscr{C}$, który powoduje zmianę własności obiektu w sposób nagły. Dalej zakładamy, że kontekst jest nieobserwowalny i nie znamy zbioru wartości $\mathscr{C}$. Dodatkowo przyjmujemy, że obserwujemy M wartości kontekstu.

W pracy rozpatrujemy dwa przypadki w zależności od charakteru obiektu:

- **przypadek deterministyczny** – obiekt jest deterministyczny; jego charakterystyka jest zależna od kontekstu oraz jest wyrażona w regułowej reprezentacji wiedzy;
- **przypadek losowy** – obiekt ma charakter losowy, tzn. jest opisany rozkładem łącznym prawdopodobieństwa wejściowych i wyjściowych zmiennych losowych; rozkład łączny jest zależny od kontekstu.

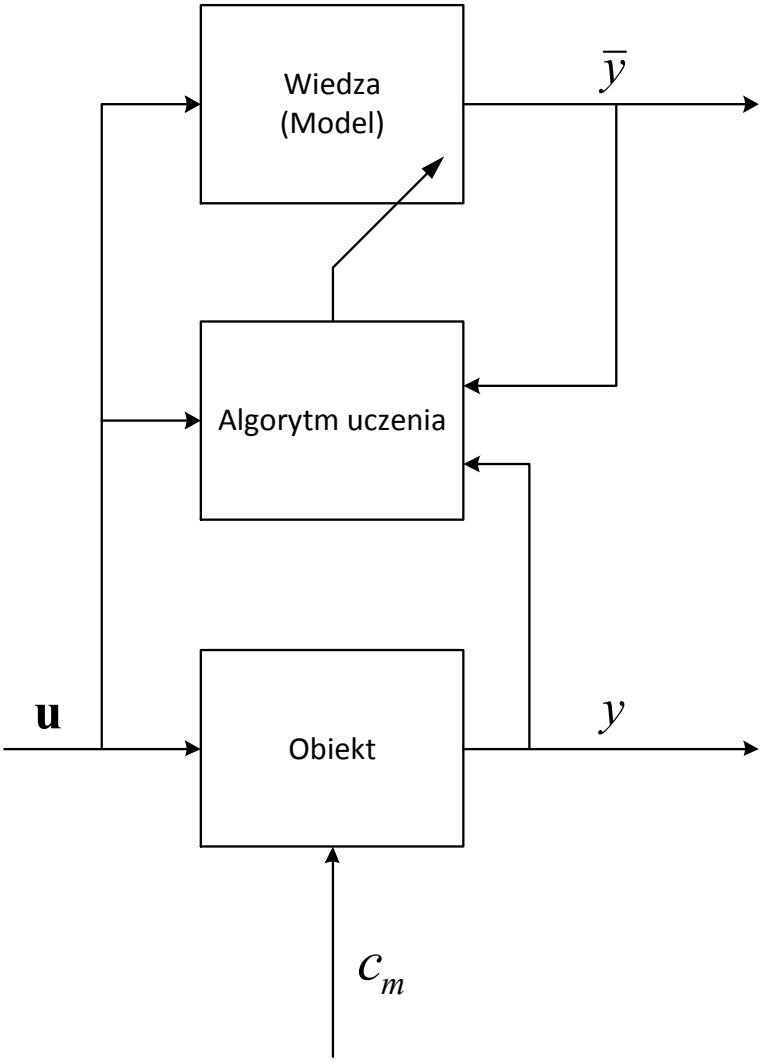
W obu przypadkach wiedza ekstrahowana z danych wyrażona jest za pomocą reguł oraz za pomocą tego samego algorytmu uczenia. Tym niemniej jej interpretacja jest odmienna w zależności od rozpatrywanego obiektu. Gdy zakładamy, że obiekt jest deterministyczny, to możemy powiedzieć, że „poznajemy" charakterystykę obiektu dla zadanego kontekstu. Innymi słowy, im więcej obserwacji posiadamy, tym lepiej „poznajemy" obiekt. Natomiast dla obiektu losowego „znajdujemy" takie wzorce, które minimalizują ryzyko w podejmowaniu decyzji. Oznacza to, że im więcej obserwacji posiadamy, tym lepsze (w sensie ryzyka) decyzje podejmujemy.

Przypadek deterministyczny

Zakładamy, że obiekt deterministyczny opisany jest za pomocą relacji (zbioru par wejść i wyjść) [27, 142, 145] zależnej od kontekstu

$$R(c_m) = \Big\{(\mathbf{u}, y) \in \mathscr{U} \times \mathscr{Y} : \varpi\Big[\varphi(\mathbf{u}, y; c_m)\Big] = 1\Big\}, \tag{1.1}$$

gdzie $\varpi[\cdot] \in \{0, 1\}$ określa wartość logiczną, $\varphi(\mathbf{u}, y; c_m)$ – charakterystyka (własność) obiektu, która jest zależna od kontekstu w m-tym momencie.



Rysunek 1.2: System ekstrakcji wiedzy o obiekcie z zaznaczeniem wpływu kontekstu.

Dla uproszczenia i przejrzystości zapisu wprowadźmy następujące oznaczenia:

$$R(c_m) \stackrel{df}{=} R_m, \tag{1.2}$$

oraz

$$\varphi(\mathbf{u}, y; c_m) \stackrel{df}{=} \varphi_m. \tag{1.3}$$

W ogólności charakterystyka obiektu może być wyrażona za pomocą jednej z reprezentacji wiedzy podanych w rozdziale 1.2.2, jednak w pracy skupiamy się na złożonych funkcjach logicznych w postaci reguł decyzyjnych.

Ze względu na złożoność obliczeniową modelowania z modelem niestacjonarnym, dalej stosujemy podejście z modelem stacjonarnym, tzn. zakładamy postać relacji aproksymującej relację (1.1),

$$\bar{R} = \Big\{(\mathbf{u}, y) \in \mathscr{U} \times \mathscr{Y} : \varpi\Big[\Phi(\mathbf{u}, y)\Big] = 1\Big\} \tag{1.4}$$

gdzie $\Phi(\mathbf{u}, y)$ oznacza model jako zestaw zdań logicznych (reguł decyzyjnych). Dla dalszej przejrzystości będziemy pisali

$$\Phi(\mathbf{u}, y) \stackrel{df}{=} \Phi. \tag{1.5}$$

Na rysunku 1.2 przedstawiono schemat ekstrakcji wiedzy z zaznaczeniem podanych wcześniej pojęć, gdzie m określa moment obserwacji kontekstu, c_m oznacza wartość kontekstu w m-tym momencie, $\mathbf{u}$ określa wartość wejść, y – wyjście takie, że $(\mathbf{u}, y) \in R_m$, $\bar{y}$ – wyjście modelu takie, że $(\mathbf{u}, \bar{y}) \in \bar{R}$, obiekt opisany jest za pomocą relacji (1.1), natomiast wiedza – relacji (1.4).

W praktyce dysponujemy ciągiem obserwacji dla zmiennego kontekstu. Ze względu na rozpatrywany przypadek deterministycznych, ciąg ten można podzielić na M kontekstów, ponieważ obaserwacje muszą być spójne na zadanych kontekstach[1]. Zatem dla każdego m-tego kontekstu dysponujemy obserwacjami (danymi):

$$\mathcal{D}_m = \{(\mathbf{u}_n, y_n) : (\mathbf{u}_n, y_n) \in R_m, n = 1, 2, \ldots, N_m\}, \tag{1.6}$$

gdzie N_m oznacza liczbę obserwacji dla m-tego kontekstu.

[1]Obserwacje są spójne, gdy dla tych samych wartości wejść obserwujemy te same wartości wyjścia.

Zadanie ekstrakcji wiedzy sprowadza się do wyznaczenia modelu dla każdego kontekstu c_m, $m = 1, 2, \ldots, M$, poprzez minimalizowanie zadanego kryterium jakości, które definiujemy w następujący sposób:

$$Q(\Phi; \mathcal{D}_m) = \sum_{(\mathbf{u},y)\in\mathcal{D}_m} \delta(\bar{y}, y), \tag{1.7}$$

gdzie Φ oznacza model, $\mathcal{D}_m$ oznacza obserwacje dla m-tego kontekstu, $\bar{y}$ jest wyjściem modelu takim, że $(\mathbf{u}, \bar{y}) \in \bar{R}$, y jest wyjściem obiektu takim, że $(\mathbf{u}, y) \in R_m$, δ jest metryką dyskretną (*delta Kroneckera*), tzn.

$$\delta(a, b) = \begin{cases} 1, & \text{jeśli } a \neq b, \\ 0, & \text{jeśli } a = b. \end{cases} \tag{1.8}$$

Metryka ta określa *błąd* między wyjściem modelu a wyjściem obiektu (błąd podejmowania decyzji), dlatego interesuje nas minimalizowanie kryterium (1.7).

Sformułowanie problemu 1.1. **Ekstrakcja wiedzy w przypadku deterministycznym**

DANE:

- ciąg uczący, tzn. obserwacje dla każdego kontekstu, $\mathcal{D}_m$, $m = 1, 2, \ldots, M$;
- klasa modeli (reprezentacja wiedzy);
- kryterium jakości Q, tj. (1.7).

SZUKANE:

- dla każdego kontekstu c_m, $m = 1, 2, \ldots, M$, model Φ_m, dla którego zadane kryterium Q przyjmuje minimalną wartość,

$$Q(\Phi_m; \mathcal{D}_m) = \min_{\Phi} Q(\Phi; \mathcal{D}_m).$$

Uwaga. Warto zauważyć, że tak określone kryterium dla każdego kontekstu jest równoważne z pojęciem *błędu klasyfikacji* [86], które jest powszechnie stosowane w zadaniu klasyfikacji i rozpoznawania.

Przypadek losowy

W przypadku losowym zakładamy, że wejścia są zmiennymi losowymi o rozkładzie prawdopodobieństwa[2] $p(\mathbf{u}|c_m)$ oraz wyjście jest zmienną losową o rozkładzie prawdopodobieństwa $p(y|\mathbf{u}, c_m)$. Rozkłady te są rozkładami niestacjonarnymi ze względu na fakt istnienia zależności od kontekstu. Dla przypadku losowego na rysunku 1.2 wiedza rozumiana jest jak relacja $\bar{R}$, natomiast obiekt opisany jest rozkładem łącznym $p(\mathbf{u}, y|c_m)$.

Dalej przyjmujemy, że dla każdego m-tego kontekstu dysponujemy obserwacjami wejść i wyjść, tj.

$$\mathcal{D}_m = \{(\mathbf{u}_n, y_n) : (\mathbf{u}_n, y_n) \sim p(\mathbf{u}, y|c_m), n = 1, 2, \ldots, N_m\} \tag{1.9}$$

gdzie N_m – liczba obserwacji dla m-tego kontekstu, symbol $\sim$ oznacza, że obserwacje są realizacjami zmiennych losowych o rozkładzie łącznym, który można wyrazić w następujący sposób

$$p(\mathbf{u}, y|c_m) = p(\mathbf{u}|c_m) \cdot p(y|\mathbf{u}, c_m). \tag{1.10}$$

Zauważmy, że obserwacje dla zadanego kontekstu są niezależne i o jednakowym rozkładzie (ang. *independent and identically distributed*, iid). Własność ta zostanie wykorzystana w zadaniu wykrywania momentów zmian kontekstu.

W zadaniu ekstrakcji wiedzy interesuje nas znalezienie modelu takiego, który dla każdego kontekstu c_m, $m = 1, 2, \ldots, M$, minimalizuje następujące kryterium (ryzyko popełnienia błędu):

$$Q_p(\Phi; c_m) = \mathbb{E}_{\mathbf{u}, y|c_m}\Big[\delta(\bar{y}, y)\Big] \tag{1.11}$$

gdzie Φ oznacza model, $\mathbb{E}$ – wartość oczekiwaną, δ jest metryką dyskretną (w teorii decyzji mówi się o *zero-jedynkowej funkcji strat*), $\bar{y}$ jest wyjściem modelu takim, że $(\mathbf{u}, \bar{y}) \in \bar{R}$, oraz y jest wyjściową zmienną losową.

[2]Zmienna losowa rozróżnia rozkład prawdopodobieństwa. Zamiast pisać $p_u(u)$ używamy $p(u)$, jednocześnie rozróżniając rozkłady $p(u)$ i $p(y)$ z powodu różnych argumentów. W literaturze często rozkład prawdopodobieństwa określa się jako *model* [13].

Sformułowanie problemu 1.2. **Ekstrakcja wiedzy w przypadku losowym**

DANE:

- ciąg uczący, tzn. obserwacje dla każdego kontekstu, $\mathcal{D}_m$, $m = 1, 2, \ldots, M$;
- klasa modeli (reprezentacja wiedzy);
- kryterium jakości Q_p, tj. (1.11);

SZUKANE:

- dla każdego $m = 1, 2, \ldots, M$, model Φ_m, który minimalizuje zadane kryterium,

$$Q_p(\Phi_m; c_m) = \min_{\Phi} Q_p(\Phi; c_m).$$

W praktyce dysponujemy ciągiem uczącym podzielonym dla każdego kontekstu (1.9), dlatego możemy jedynie wyznaczyć empiryczne przybliżenie kryterium (1.11). Zakładając, że ciąg obserwacji jest podzielony na M kontekstów, analogicznie jak to było w przypadku deterministycznym, wówczas zazwyczaj stosuje się metodę indukcyjną minimalizowania kryterium empirycznego (ang. *Empirical Risk Minimization*, ERM) [21, 156], która sprowadza się do rozpatrzenia następującego kryterium dla każdego $\mathcal{D}_m$, $m = 1, 2, \ldots, M$:

$$\hat{Q}_p(\Phi; \mathcal{D}_m) = \sum_{(\mathbf{u}, y) \in \mathcal{D}_m} \delta(\bar{y}, y). \tag{1.12}$$

W celu stosowania metody minimalizowania kryterium empirycznego należy wyznaczyć zmiany kontekstu. Zadanie to w literaturze określa się jako *wykrywanie zmian* (ang. *change detection*) i definiuje w następujący sposób [10, 58]:

Sformułowanie problemu 1.3. **Wykrywanie momentów zmian kontekstu**

DANE:

- próby z rozkładów prawdopodobieństwa $p(\mathbf{u}, y|c_{m-1})$ i $p(\mathbf{u}, y|c_m)$;

- miara niepodobieństwa rozkładów prawdopodobieństwa ϱ;
- wartość parametru wrażliwości, $\sigma > 0$.

SZUKANE:

- momenty takie, że:

$$\tau = \Big\{ m : \varrho\Big(p(\mathbf{u}, y|c_{m-1}), p(\mathbf{u}, y|c_m)\Big) \geqslant \sigma \Big\}.$$

Alternatywnym podejściem do minimalizowania empirycznego kryterium dla każdego kontekstu jest rozpatrywanie innego, łącznego kryterium dla wszystkich kontekstów, zwanego *predykcyjnym błędem sekwencyjnym* (ang. *predictive sequential error* lub *prequential error*) [48]. Jeżeli założymy, że dane napływają pojedynczo w strumieniu i są numerowane wg momentu pojawienia się (z zachowaniem kolejności kontekstów), tj. dysponujemy ciągiem uczącym

$$\mathcal{D} = \{(\mathbf{u}_n, y_n) : n = 1, 2, \ldots, N\}, \tag{1.13}$$

oraz na podstawie wiedzy dokonywana jest predykcja wyjścia na podstawie wartości wejściowych, to wówczas rozpatrujemy

$$\overline{Q}(\mathbf{\Phi}, \mathcal{D}) = \frac{1}{N} \sum_{n=1}^{N} \delta(\bar{y}_n, y_n), \tag{1.14}$$

gdzie $\mathbf{\Phi} = (\Phi_1, \Phi_2, \ldots, \Phi_N)$ jest ciągiem modeli.

W podejściu z minimalizowaniem predykcyjnego błędu sekwencyjnego zakłada się stosowanie uczenia przyrostowego z zapominaniem, ponieważ wyznaczany jest ciąg modeli, w którym modele są w każdym momencie walidowane i uaktualniane. Innymi słowy, wiedza ekstrahowana jest na bieżąco i bez stosowania wykrywania momentów zmian.

Algorytmy ekstrakcji wiedzy

Zarówno w przypadku deterministycznym i losowym, dla przyjętego podejścia z modelem stacjonarnym, stosuje się dwa warianty uczenia przyrostowego z zapominaniem [38, 96, 150]:

Wariant 1: Podejście **z wyznaczaniem zmian kontekstu**, czyli ciąg uczący dzielony jest na M kontekstów. W przypadku deterministycznym można tego dokonać poprzez sprawdzanie spójności obserwacji, natomiast w przypadku losowym poprzez zastosowanie metody wykrywania zmian. Wówczas, dla każdego c_m, $m = 1, 2, \ldots, M$, wyznaczamy:

$$\Phi_m := G_1\Big(\mathcal{D}_m\Big), \tag{1.15}$$

gdzie G_1 oznacza algorytm uczenia przyrostowego, zwanego też *tymczasowym uczeniem wsadowym* (ang. *temporal batch learning*) [98], $\mathcal{D}_m$ – obserwacje dla m-tego kontekstu.

Wariant 2: Podejścia **z uaktualnianiem modelu** (bez wykrywania zmian) przy użyciu obserwacji zawartych w oknie przesuwnym[3]:

$$\Phi_{N+1} := G_2\Big(\Phi_N, \mathcal{D}_{N+1}^L\Big), \tag{1.16}$$

gdzie G_2 oznacza algorytm uczenia przyrostowego z oknem przesuwnym (ang. *learning with shifting window*), $\mathcal{D}_{N+1}^L$ – okno przesuwne, lub w oparciu o pojedynczą obserwację, tzn.

$$\Phi_{N+1} := G_3\Big(\Phi_N, \{(\mathbf{u}_{N+1}, y_{N+1})\}\Big), \tag{1.17}$$

gdzie G_3 oznacza algorytm uczenia przyrostowego *ze strojonym modelem* (ang. *learning with self-adjusting model*) [21].

Uwaga1. W celu wyznaczenia modeli w wariancie 1 korzystamy z kryterium (1.7) lub (1.12), natomiast w wariancie 2 – (1.14).

Uwaga 2. Zwróćmy uwagę na różnicę dwóch wariantów uczenia przyrostowego z zapominaniem. Pierwsze podejście jest *retrospektywne*, czyli wiedza ekstrahowana jest **po** określeniu momentu zmiany kontekstu. Natomiast drugie podejście jest *prospektywne*, tzn. interesuje nas posiadanie bieżącej wiedzy. Zatem jeżeli istnieje potrzeba posiadania wiedzy dla całego kontekstu, to preferowane jest tymczasowe uczenie wsadowe z zastosowaniem

[3]Okno przesuwne oznacza ciąg L ostatnich obserwacji licząc od bieżącego momentu obserwacji $N+1$, tj. $\mathcal{D}_{N+1}^L = \{(\mathbf{u}_n, y_n) : n = N+2-L, N+3-L, \ldots, N, N+1\}$. Indeks dolny oznacza bieżący moment (koniec okna), natomiast indeks górny – liczbę obserwacji w oknie przesuwnym.

wykrywania zmian kontekstu. Natomiast w przypadku, gdy istotne jest podejmowanie decyzji na bieżąco, to rozsądniejszym rozwiązaniem jest uczenie z mechanizmem zapominania (z oknem przesuwnym lub ze strojonym modelem).

Na koniec zauważmy, że w wariancie 1 najważniejsze jest określenie mechanizmu wykrywania zmian. Po określeniu momentów zmian kontekstu można stosować dowolny algorytm z uczeniem wsadowym w celu uzyskania wiedzy regułowej. Dlatego też dla tego przypadku podany zostanie ogólny schemat algorytmu, natomiast główny nacisk zostanie położony na metodzie wykrywania zmian.

1.3 Aktualny stan badań

Ekstrakcja wiedzy w przypadku stacjonarnym (brak zmiennego kontekstu) jest problemem dobrze znanym i omówionym w literaturze przedmiotu [34, 35, 91, 103, 106]. Zaproponowano metody, które skutecznie rozwiązują problem indukcji reguł, m.in. algorytm AQ (ang. *Algorithm Quasi-Optimal*) [102], algorytm CEA (ang. *Candidate Elimination Algorithm*) [106], algorytm CN2 (od nazwisk twórców – Clarka i Nibletta, wersja 2) [36]. Ponadto stosuje się również inne podejścia, m.in. w oparciu o drzewa decyzyjne, np. algorytm ID3 (ang. *Induction of Decision Trees*) [123], o teorię obliczeń i teorię probabilistycznego uczenia przybliżonego PAC (ang. *Probably Approximately Correct learning*) [3, 61, 155], o teorię statystycznego uczenia [128], o teorię zbiorów przybliżonych w postaci grafowej (grafy przepływów, ang. *flow graphs*) [117, 118] oraz teorię grafów, np. zastosowanie grafowej przestrzeni przeszukiwań do predykcji [62], wyznaczania grup [52], czy znajdowania wzorców [64, 66, 121, 159].

Kolejnym zagadnieniem rozpatrywanym w ekstrakcji wiedzy regułowej jest możliwość przetwarzania obserwacji napływających w strumieniu danych. Problem ten zazwyczaj rozwiązuje się za pomocą modyfikacji algorytmów z uczeniem wsadowym, tak jak np. w algorytmie AQ-11 (modyfikacja algorytmu AQ) [105], czy w algorytmie GEM (ang. *Generalization of Examples by Machine*, rozwinięcie algorytmu AQ) [127], oraz ID5 (rozwinięcie metody ID3) [154], YAILS (dopuszczenie nadmiarowości reguł w algorytmie AQ) [151].

W przypadku z ukrytym kontekstem dotychczas zaproponowano dwie grupy algoryt-

mów rozwiązujących zagadnienie indukcji reguł dla obiektów niestacjonarnych.

W podejściu z tymczasowym uczeniem wsadowym kluczowym elementem jest metoda wykrywania zmian kontekstu. Jednym z proponowanych rozwiązań jest wykrywanie zmian kontekstu w oparciu o sprawdzanie liczby błędów popełnianych przez klasyfikator. Gdy liczba błędów zaczyna rosnąć, to zgłaszana jest zmiana. Taki schemat zaproponowano m.in. w algorytmie WAH (ang. *Window Adjustment Heuristics*) [162], metodzie DDM (ang. *Drift Detection Method*) [50] oraz jej modyfikacji EDDM (ang. *Early Drift Detection Method*) [8], SPLICE z indukcją drzew decyzyjnych [60], ACDD (ang. *Adaptive Concept Drift Detection*) [42]. Ostatnia z wymienionych metod korzysta z własności dotyczących ograniczenia na tempo zmian kontekstu wynikających z teorii obliczeń [9].

Odmiennym podejście do wykrywania zmian kontekstu jest zastosowanie wnioskowania statystycznego. Część metod sprowadza się do porównywania rozkładów prawdopodobieństw za pomocą zadanej miary niepodobieństwa. Ze względu na fakt posiadania jedynie obserwacji, rozkłady są szacowane za pomocą danych zawartych w dwóch sąsiadujących oknach przesuwnych [141]. Stosuje się różne miary niepodobieństwa, m.in. normę L_1 [77], entropię [141, 158], miarę Kullbacka-Leiblera [136]. Innym rozwiązaniem jest wykorzystanie testów statystycznych i funkcji wiarygodności [10, 58]. Ponadto rozróżnia się podejście parametryczne do szacowania rozkładów prawdopodobieństwa (stosowanie zadanych postaci parametrycznych rozkładów, np. [10, 58]) i nieparametryczne poprzez wykorzystanie np. histogramów [136], czy procesów Gaussa [126]. Kolejnym podejściem jest zastosowanie modelowania bayesowskiego, w którym zakłada się, że liczba zmian oraz momenty zmian są zmiennymi losowymi (tzw. *procesy punktowe*, [45]) lub czas pomiędzy zmianami kontekstów jest zmienną losową [1].

Ze względu na fakt, iż tymczasowe uczenie przyrostowe nie może być stosowane do bieżącej predykcji i podejmowania decyzji, dlatego też zaproponowano rozwiązania problemu indukcji reguł w oparciu o uczenie z uaktualnianiem. Jako mechanizm zapominania w algorytmie ekstrakcji wiedzy stosuje się zazwyczaj okno przesuwne, m.in. w algorytmach FLORA (ang. *FLOating Rough Approximation*) [84, 162, 163], AQ-PM (ang. *Algorithm Quasi-Optimal with Partial Memory*), FACIL (ang. *Fast and Adaptive Classiffer by Incremental Learning*, tylko dla atrybutów numerycznych) [46], metodzie indukcji drzew CVFDT (ang. *Concept-adapting Very Fast Decision Tree Learner*) [65], indukcji reguł z grafu informacyjno-

rozmytego OLIN (ang. *On-Line Information Network*) [88, 89]. Dodatkowo w algorytmach AQ-PM, FLORA i FACIL stosuje się zapominanie wybiórcze, podobnie jak w metodzie STAGGER (STorage AGGregation Evaluation Refinement) [134]. Jednak w każdym z wymienionych przypadków mechanizm zapominania wymaga przechowywania części obserwacji.

Ze względu na wciąż niezadowalające rezultaty proponowanych algorytmów, zadanie ekstrakcji wiedzy dla ukrytego kontekstu wskazuje się jako jeden z 10 najważniejszych problemów badawczych w dziedzinie eksploracji danych i uczenia maszynowego [166].

1.4 Cel i zakres pracy

Dotychczas dla potrzeb ekstrakcji wiedzy regułowej o obiektach niestacjonarnych opracowano algorytmy uczenia, które nie dają satysfakcjonujących wyników [166]. Korzystając z przedstawionego problemu ekstrakcji wiedzy w poprzednich punktach pracy możemy sformułować cel pracy.

Celem pracy jest opracowanie algorytmów uczenia przyrostowego z zapominaniem dla ekstrakcji wiedzy wyrażonej za pomocą reguł decyzyjnych dla obiektów niestacjonarnych, które pozwalają na analizę retrospektywną (tymczasowe uczenie wsadowe) oraz prospektywną (uczenie z oknem przesuwnym i strojonym modelem). Aby osiągnąć zamierzony cel należy rozwiązać następujące zadania:

1. Opracować metody wykrywania zmian kontekstu dla tymczasowego uczenia wsadowego.
2. Opracować metody ekstrakcji wiedzy z oknem przesuwnym.
3. Opracować metodę ekstrakcji wiedzy ze strojonym modelem.

W **zakres pracy** wchodzą następujące elementy:

1. Opracowanie algorytmów wykrywania zmian kontekstu z użyciem:

- modelowania częstościowego – szacowanie rozkładów prawdopodobieństwa za pomocą histogramów i zastosowaniem miary niepodobieństwa rozkładów;
- modelowania bayesowskiego – wykorzystanie rozkładów prawdopodobieństwa zmiennych dyskretnych oraz aproksymacji współczynnika Bayesa;

dla tymczasowego uczenia wsadowego w celu retrospektywnej analizy obiektu.

2. Opracowanie algorytmu uczenia przyrostowego wykorzystującego mechanizm zapominania z oknem przesuwnym w celu prospektywnej analizy obiektu.

3. Opracowanie algorytmu uczenia przyrostowego wykorzystującego mechanizm zapominania z oknem przesuwnym oraz zapominaniem wybiórczym w celu umożliwienia prospektywnej analizy obiektu.

4. Opracowanie algorytmu uczenia przyrostowego ze strojonym modelem, wykorzystującego reprezentacje grafowe do:

   - agregacji obserwacji;
   - regularyzacji klasy modeli regułowych;
   - ograniczeniu przestrzeni przeszukiwań reguł,

   w celu umożliwienia prospektywnej analizy obiektu.

5. Przeprowadzenie badań symulacyjnych oraz empirycznych mających na celu zweryfikowanie poprawności i skuteczności działania proponowanych algorytmów ekstrakcji wiedzy i wykrywania zmian w porównaniu z metodami znanymi w literaturze.

Prezentowana praca poszerza aktualny stan wiedzy i zakres dostępnych technik w dziedzinie uczenia maszynowego ze szczególnym uwzględnieniem reprezentacji regułowej. Rezultaty pracy będą przydatne do opracowywania komputerowych systemów wspomagania podejmowania decyzji.

W pracy stawiana jest następująca **teza**:
„Zastosowanie uczenia przyrostowego z zapominaniem dla ekstrakcji reguł decyzyjnych pozwala na posiadanie aktualnej wiedzy o obiekcie niestacjonarnym, tj. obiekcie, którego własność zależna jest od zmiennego kontekstu."

1.5 Układ pracy

Rozprawa składa się z niniejszego rozdziału oraz sześciu kolejnych.

Rozdział 2. Scharakteryzowano regułową (logiczną) reprezentację wiedzy.

Rozdział 3. Zaproponowano dwie metody wykrywania zmian kontekstu. Szczegółowo opisano podejście z zastosowaniem modelowania częstościowego oraz bayesowskiego.

Rozdział 4. Zaproponowano dwa algorytmy ekstrakcji wiedzy regułowej z oknem przesuwnym, które są modyfikacjami algorytmu AQ.

Rozdział 5. Zaproponowano algorytm ekstrakcji wiedzy z uaktualnianiem poprzez wykorzystanie reprezentacji grafowych.

Rozdział 6. Przedstawiono wyniki badań empirycznych. Działanie proponowanych metod wykrywania zmian kontekstu porównano z algorytmami znanymi w literaturze przedmiotu na podstawie benchmarkowego zbioru danych oraz przedstawiono zastosowanie do systemów zorientowanych na usługi. Działanie proponowanych algorytmów ekstrakcji wiedzy regułowej porównano z algorytmami znanymi w literaturze przedmiotu na podstawie benchmarkowych zbiorów danych w przypadku deterministycznym oraz losowym. Ponadto zaprezentowano zastosowanie w systemie wspomagania leczenia terapii cukrzycy.

Rozdział 7. Podano uwagi końcowe ze wskazaniem nowości prezentowanej pracy oraz wskazaniem proponowanych dalszych kierunków badań.

Rozdział 2

Regułowa reprezentacja wiedzy

2.1 Definicje i oznaczenia

W niniejszej pracy rozpatrujemy wejścia (zwanymi też *atrybutami* lub *cechami*) oraz wyjścia, które mają charakter dyskretny. Model jest zestawem zdań logicznych, które nazywa się regułami. Przestrzeń modeli regułowych, oznaczany przez $\mathscr{F}$, reprezentowany jest za pomocą *logiki z atrybutami* (ang. *Attribute-Value Logic*) [22, 23, 93, 102]. Wyszczególnia się w niej *formuły elementarne* (zwane też *własnościami elementarnymi* [72]), które dotyczą wejścia i wyjścia:

- formułę elementarną $\alpha_k^d =$ "$u^d = k$", gdzie $k \in \mathscr{U}_d$, nazywamy **wejściową** i odczytujemy w następujący sposób: d-te wejście przyjmuje wartość równą k_d;
- formułę elementarną $\alpha_l^{out} =$ "$y = l$", gdzie $l \in \mathscr{Y}$, nazywamy **wyjściową** i odczytujemy w następujący sposób: wyjście przyjmuje wartość równą l.

W klasycznym rachunku zdań formuły elementarne odpowiadają zdaniom logicznym [125], które mają interpretację określoną jak wyżej. Wartość logiczna formuły elementarnej α określa, czy formuła jest prawdziwa (w sensie logicznym), tj. $\varpi[\alpha] = 1$, czy fałszywa, tj. $\varpi[\alpha] = 0$.

Dla rozpatrywanego przypadku z D wejściami mamy K wejściowych formuł elementarnych oraz Y wyjściowych formuł elementarnych. Zbiór wszystkich formuł elementarnych oznaczmy przez $\mathscr{A}$, $\mathrm{card}\{\mathscr{A}\} = K + Y$.

Ponadto w logice z atrybutami zakłada się konkretną postać wyrażeń logicznych, tzn. dopuszcza się *operatory logiczne* takie jak [125]: *i* (*koniunkcja*) – $\wedge$, *lub* (*dysjunkcja*) – $\vee$, *jeśli ... to ...* – $\Rightarrow$. Znak równoważności nie jest wykorzystywany.

Reguły (zwane też *regułami decyzyjnymi*, *regułami klasyfikacyjnymi*, *regułami produktowymi*) wyraża się w następującej postaci:

JEŚLI *warunek*, TO *decyzja*

gdzie *warunek* jest koniunkcją formuł elementarnych wejściowych, czyli jest wyrażeniem logicznym w *1-koniunkcyjnej postaci normalnej* (1-CNF), natomiast *decyzja* jest pojedynczą wyjściową formułą elementarną. Zatem reguła ϕ przedstawiona jest w następujący sposób:

$$\phi = \Big(\phi_{in} \Rightarrow \phi_{out}\Big), \tag{2.1}$$

gdzie lewa część implikacji oznacza warunek, $\phi_{in} = \bigwedge\limits_{d\in\mathscr{D}} \alpha^d_{k_d}$, $\mathscr{D} \subseteq \{1, 2, \ldots, D\}$, $\alpha^d_{k_d}$ jest wybraną formułą elementarną w d-tym wejściu o wartości równej $k_d \in \mathscr{U}_d$, natomiast prawa strona określa decyzję, $\phi_{out} = \alpha^{out}_l$.

Model (wiedza) i charakterystyka obiektu w przypadku deterministycznym są zestawem zdań logicznych w postaci (2.1), które są połączone spójnikiem logicznym *lub*. Innymi słowy, reguły dla każdej wartości wyjścia modelu są wyrażone w k-dysjunkcyjnej postaci normalnej (k-DNF; w rozpatrywanym przypadku $k = D$), tzn. że wyrażenia 1-CNF, które zawierają co najwyżej D koniunkcji alternatyw, tj. tyle, ile jest wejść, połączone są operatorami dysjunkcji [15, 79].

Podane powyżej pojęcia zobrazowano na następującym przykładzie.

Przykład 2.1. *Dany jest obiekt o dwóch wejściach, $u^1 \in \{a, b\}$ i $u^2 \in \{1, 2\}$, oraz wyjściu $y \in \{0, 1\}$. Liczności zbiorów wartości wejść wynoszą odpowiednio $K_1 = 2$, $K_2 = 2$, zatem liczba wejściowych formuł elementarnych wynosi $K = 4$. Wyjściowych formuł elementarnych jest $Y = 2$. Natomiast zbiór wszystkich (zarówno wejściowych, jak i wyjściowych) formuł elementarnych jest następujący:*

$$\mathscr{A} = \{\alpha^1_a, \alpha^1_b, \alpha^2_1, \alpha^2_2, \alpha^{out}_0, \alpha^{out}_1\}.$$

Zbiór reguł z warunkiem w postaci 1-CNF zbudowanych z wejściowych formuł elementarnych z $\mathscr{A}$ oraz $Y = 2$ zawiera 2^K reguł [110].

Przykładowa charakterystyka obiektu dla ustalonego kontekstu c_n może być postaci:

$$\varphi_m = \phi_1 \vee \phi_2 \vee \phi_3$$

gdzie

$$\phi_1 = \left(\alpha_a^1 \wedge \alpha_1^2 \Rightarrow \alpha_1^{out}\right),$$
$$\phi_2 = \left(\alpha_b^1 \Rightarrow \alpha_0^{out}\right),$$
$$\phi_3 = \left(\alpha_2^2 \Rightarrow \alpha_0^{out}\right).$$

■

2.2 Własności regułowej reprezentacji wiedzy

Regułowa reprezentacja wiedzy jest jedną z najstarszych reprezentacji wiedzy wykorzystywanych w sztucznej inteligencji i uczeniu maszynowym [34]. Stosowanie operatorów logicznych oraz zdefiniowanie formuł elementarnych pozwala na łatwe wyrażanie i zrozumienie pojęć w języku naturalnym oraz dostarcza uniwersalnych zasad wnioskowania.

Można podać następujące własności, które wyszczególniają logiczną reprezentację wiedzy spośród innych reprezentacji [79]:

- łatwość interpretacji wiedzy przez człowieka;
- łatwość automatycznej translacji wiedzy do sformułowania w języku naturalnym;
- łatwość modyfikacji wiedzy przez człowieka lub system ekspertowy;
- łatwość wykorzystania wiedzy w systemach ekspertowych;
- łatwość interpretacji zjawisk wielowymiarowych;
- dobra skuteczność jako model charakteryzujący [103] i dyskryminacyjny [79];
- uniwersalne zasady wnioskowania (zasady rozumowania dedukcyjnego [125], metoda logiczno-algebraiczna [22, 23, 27].

Regułowa reprezentacja wiedzy sprawdza się przede wszystkim tam, gdzie następuje interakcja człowieka z maszyną. Szczególnie, gdy człowiek potrzebuje szybkiej analizy zjawiska i dodatkowo może nie posiadać umiejętności posługiwania się innymi modelami, np. probabilistycznymi. Ponadto, model regułowy nie nastręcza problemów w interpretacji zjawisk wielowymiarowych, ponieważ konstrukcja *warunku* daje natychmiastową możliwość zrozumienia procesu ze względu na lokalną niezależność wszystkich wymiarów.

Natomiast z technicznego punktu widzenia regułowa reprezentacja wiedzy jest kusząca ze względu na łatwość przekształcania jej do wyrażeń w języku naturalnym. Poza tym umożliwia łączenie wiedzy z różnych źródeł oraz usuwanie ewentualnych konfliktów [112]. Dlatego też wiele systemów ekspertowych opartych było i jest na logicznych reprezentacjach wiedzy [22, 68, 72].

Z podanych wyżej przyczyn regułowa reprezentacja wiedzy znalazła liczne zastosowania w procesach podejmowania decyzji, m.in. w medycynie i biologii [11, 19, 51, 90, 148], w procesach przemysłowych [87], w ekonomii i finansach [44, 87], w wykrywaniu ataków sieciowych [97, 99], w zarządzaniu obciążeniem w sieci [85], w analizie zachowań klientów telekomunikacyjnych [14], w analizie zachowań użytkowników systemów informatycznych [30, 81, 82, 160].

Stosowanie wiedzy regułowej w podejmowaniu decyzji jest podejściem dyskryminacyjnym, w odróżnieniu od podejścia generującego [13]. Oznacza to, że takie podejście pozwala, dla zadanego wejścia, na określenie wyjścia. Natomiast wygenerowanie zbioru wejść oraz wyjść jest niemożliwe.

Jednak oprócz wielu zalet, regułowa reprezentacja wiedzy posiada również wady, do których można zaliczyć [79]:

- wysoki wymiar Vapnika-Chervonenkisa[1];
- zbytnie dopasowanie się modelu do danych (ang. *overfitting*);
- indukcja reguł jest problemem NP-zupełnym [2].

[1]Wymiar Vapnika-Chervonenkisa (*VC-dim*) określa pewną *pojemność* algorytmu klasyfikacji lub jego zdolność do *generalizacji*. *VC-dim* określany jest jako liczność przynajmniej jednego, największego podzbioru przestrzeni konceptów, dla którego klasyfikator może dokonać dowolnej dychotomii tego podzbioru. Formalną definicję można znaleźć w [15, 31, 34, 62, 83, 156, 157].

O ile wymiar Vapnika-Chervonenkisa dla $1\text{-}CNF$ jest niewielki, tj. *VC-dim* $= K$ [110], to dla klasy $k\text{-}DNF$ już tak nie jest, tzn. *VC-dim* $= \prod_{d=1}^{D} K_d$ [2]. Fakt ten implikuje, że występuje groźba zbytniego dopasowania się modelu do danych.

Ponadto, duża wartość wymiaru Vapnika-Chervonenkisa pociąga za sobą potrzebę posiadania dużej liczby obserwacji w celu poprawnego przeprowadzenia procesu uczenia, co wynika z następującego twierdzenia (kryterium jakości jak (1.11), Φ – model) [39]:

Twierdzenie 2.1. *(Vapnik-Chervonenkis)*
Załóżmy, że $\mathrm{card}\{\mathscr{Y}\} = 2$, $\mathrm{card}\{\mathscr{F}\} < \infty$, *oraz* $\min_{\Phi\in\mathcal{F}}\{Q_p(\Phi)\} = 0$. *Wówczas dla każdego* N *(liczby obserwacji) oraz* $\epsilon > 0$ *zachodzą wyrażenia (przez* Φ_N^* *oznaczamy najlepszy model dla* N *obserwacji)*

$$\Pr\{Q_p(\Phi_N^*) > \epsilon\} \leqslant \mathrm{card}\{\mathscr{F}\} \cdot \exp(-N \cdot \epsilon)$$

oraz

$$\mathrm{E}[Q_p(\Phi_N^*)] \leqslant \frac{1 + \log_2(\mathrm{card}\{\mathscr{F}\})}{N}.$$

Z twierdzenia tego można wyciągnąć następujący wniosek [39]:

Wniosek 2.1. *Dla* $\mathrm{card}\{\mathscr{Y}\} = 2$ *i klasy* $1\text{-}CNF$*, czyli* $\mathrm{card}\{\mathscr{F}\} = log_2 2^K = K$*, mamy:*

$$\mathrm{E}[Q_p(\Phi_N^*)] \leqslant \frac{1 + K}{N},$$

czyli dla $N > K$ *model* Φ_N^* *ma mały błąd w sensie średnim. Podobnie dla klasy* $k\text{-}DNF$*, gdzie* $log_2 2^{(\prod_{d=1}^{D} K_d)} = \prod_{d=1}^{D} K_d$*, mamy:*

$$\mathrm{E}[Q_p(\Phi_N^*)] \leqslant \frac{1 + \prod_{d=1}^{D} K_d}{N},$$

czyli dla $N > \prod_{d=1}^{D} K_d$ *model* Φ_N^* *ma mały błąd w sensie średnim.*

Zauważmy, że dla dużych rozmiarów problemów, tj. dużych wartości K i D, przy źle dobranym ciągu uczącym, tj. niedostatecznie dużym, istnieje ryzyko, że model regułowy (wyrażony w $k\text{-}DNF$) nie zostanie poprawnie wyznaczony.

Natomiast wada dotycząca nieparametryczności regułowej reprezentacji wiedzy łączy się z trudnością zastosowania uczenia przyrostowego. Dlatego też stosuje się jedynie mechanizm z oknem przesuwnym i zapominanie wprost. Wykorzystanie metody z zapominaniem wykładniczym oraz nie wprost jest niezwykle trudne (lub wręcz niemożliwe) bez zaproponowania jakichkolwiek form parametryzacji modelu.

2.3 Wiedza regułowa w zadaniu klasyfikacji

Zadanie polegające na przydzieleniu obserwacji do klasy (zwanej też decyzją) określamy mianem *zadania klasyfikacji* lub *predykcji* [27, 86]. W przypadku reguł polega ono na dopasowaniu warunkowi odpowiedniej decyzji.

Zadanie klasyfikacji z wykorzystaniem wiedzy regułowej można rozwiązać korzystając z jednej z podanych *metod podejmowania decyzyjnymi*:

1. Standardowa technika klasyfikacji z wykorzystaniem wiedzy regułowej polega na znalezieniu reguły, której warunek dokładnie pokrywa się z obserwacją i wówczas zwracana jest decyzja. Aby znaleźć pokrywające obserwację warunki można stosować przegląd wszystkich reguł lub wydajniejsze (w sensie złożoności obliczeniowej) algorytmy, jak np. metodę logiczno-algebraiczną [22, 23, 27].

2. Czasem może zdarzyć się, że w modelu nie ma żadnego warunku, który odpowiadałaby obserwacji. Wówczas można stosować *najlepsze dopasowanie* (zwane też *elastycznym dopasowaniem*) warunku do obserwacji [151]. Reguła, której warunek najlepiej wg zadanego kryterium odpowiada obserwacji, jest wybierana do podjęcia decyzji.

3. Kolejne podejście do klasyfikacji z wykorzystaniem wiedzy regułowej zakłada, że każdej regule przyporządkowywana jest waga [22, 68, 86]. Waga ta może przyjmować interpretację wskaźnika *pewności* reguły [22, 27, 68, 86], tzn. na ile dana reguła jest pewna. Wówczas sprawdzane jest pokrycie obserwacji przez warunki i decyzja jest podejmowana na podstawie reguły, która pokrywa obserwację oraz posiada najwyższą wartość wskaźnika pewności.

Rozdział 3

Ekstrakcja wiedzy z wykrywaniem zmian kontekstu

Rozdział zawiera oryginalne rezultaty pracy, tzn. dwa algorytmy wykrywania zmian kontekstu dla tymczasowego uczenia wsadowego.

3.1 Wprowadzenie

Ekstrakcja wiedzy z tymczasowym uczeniem przyrostowym składa się z dwóch kroków, tj. wykrywania zmian kontekstu oraz ekstrakcji wiedzy na horyzoncie obserwacji, na którym kontekst jest stały. Ogólny schemat algorytmu z tymczasowym uczeniem przyrostowym przedstawia procedura 3.1.1.

Algorytm 3.1.1. Algorytm ekstrakcji wiedzy z tymczasowym uczeniem wsadowym.

Wejście: (i) ciąg uczący $\mathcal{D}$, (ii) $N := 0$, (iii) $m := 1$, (iv) $\mathcal{D}_1 := \emptyset$, (v) algorytm wykrywania momentów zmian, (vi) $g(\cdot, \cdot)$ – algorytm indukcji reguł.

Wyjście: Zestaw reguł Φ_m dla każdego wykrytego kontekstu.

Krok 1: Ustaw $N := N + 1$. Jeśli $N > \mathrm{card}\{\mathcal{D}\}$, to $\Phi_m := g(\mathcal{D}_m)$ i STOP.
W przeciwnym razie pobierz obserwację $(\mathbf{u}_N, y_N)$.

Krok 2: (`Wykrywanie zmian`) Sprawdź, czy zaszła zmiana. Jeśli nie, to

$$\mathcal{D}_m := \mathcal{D}_m \cup \{(\mathbf{u}_N, y_N)\}$$

i idź do kroku 1.

Krok 3: (`Ekstrakcja wiedzy`) Ekstrahuj reguły na podstawie ciągu obserwacji $\mathcal{D}_m$, tj.

$$\Phi_m := g(\mathcal{D}_m).$$

Ustaw $m := m + 1$, $\mathcal{D}_m := \emptyset$ i idź do kroku 1.

Ekstrakcja wiedzy w powyższym algorytmie odbywa się za pomocą dowolnego wybranego algorytmu indukcji reguł. Kluczowym elementem podejścia z tymczasowym uczeniem wsadowym jest metoda wykrywania zmian kontekstu, dlatego w dalszych rozważaniach skupimy się wyłącznie na tym zagadnieniu. Rozważone są dwa podejścia. Pierwsze opiera się na modelowaniu częstościowym, natomiast drugie na modelowaniu bayesowskim[1].

3.2 Problem wykrywania zmian kontekstu

Problem wykrywania zmian kontekstu polega na znalezieniu momentów takich, że

$$\tau = \Big\{ m : \varrho\Big(p(\mathbf{u}, y|c_{m-1}), p(\mathbf{u}, y|c_m)\Big) \geqslant \sigma \Big\},$$

gdzie $\varrho : \mathscr{P} \times \mathscr{P} \to [0, \infty)$ jest miarą niepodobieństwa, $\mathscr{P}$ oznacza przestrzeń rozkładów, $\sigma > 0$ jest parametrem wrażliwości, oraz ϱ spełnia warunki $\varrho(p_1, p_2) = 0 \Leftrightarrow p_1 \equiv p_2$, $\varrho(p_1, p_2) > 0$ – w przeciwnym przypadku. W praktyce dysponujemy nie prawdopodobieństwami, ale próbami z rozkładów, dlatego w celu porównania rozkładów należy zaproponować sposób ich szacowania.

Przed przystąpieniem do sformułowania metod wykrywania zmian kontekstu dokonajmy następujących spostrzeżeń:

[1]Modelowanie bayesowskie **nie** oznacza stosowania estymatora Bayesa, ale zastosowanie bayesowskiego porównania modeli (ang. *Bayesian model comparison*) i wiarygodności modeli (ang. *model evidence* lub *marginalized likelihood*) [12, 13].

1. Wiedza regułowa jest modelem dyskryminacyjnym, tzn. w indukcji reguł istotne jest minimalizowanie błędu podejmowania decyzji. Dlatego w wykrywaniu zmian kontekstu wystarczy rozważać rozkład warunkowy wyjścia[2], $p(y|\mathbf{u}, c_m)$, nie zaś łączny rozkład zmiennych wejściowych i wyjściowych. Odpowiada to sytuacji, gdy każda reguła rozpatrywana jest osobno [6]. Określenie momentów zmian wejść nie wpływa na popełnianie błędów w stosowaniu wiedzy. Zatem interesuje nas znalezienie momentów zmian takich, że
$$\tau = \Big\{ m : \varrho\Big(p(y|\mathbf{u}, c_{m-1}), p(y|\mathbf{u}, c_m)\Big) \geqslant \sigma \Big\}.$$

2. W niniejszej pracy zakładamy, że wszystkie wejścia mają charakter dyskretny (nominalny) oraz wyjście jest dyskretne. W takim przypadku uzasadnionym rozwiązaniem w szacowaniu rozkładów zmiennych dyskretnych jest zastosowanie histogramów, jeśli tylko łączna liczba wartości wejść i wyjść nie jest zbyt duża.

3. W modelowaniu częstościowym przyjmujemy, że szacowanie prawdopodobieństw odbywa się przy pomocy dwóch sąsiadujących okien przesuwnych, na podstawie których wyznaczane są odpowiednie histogramy. Takie podejście jest uważane za odpowiednie dla metod wykrywania zmian przy użyciu miary niepodobieństwa [58, 141]. Natomiast w modelowaniu bayesowskim stosujemy pojedyncze okno przesuwne, na którym dokonujemy wyboru między modelem uwzględniającym zmianę kontekstu oraz takim, który tej zmiany nie uwzględnia.

3.3 Podejście częstościowe

W modelowaniu częstościowym zakłada się, że rozkład prawdopodobieństwa opisujący rozpatrywany obiekt jest jednoznacznie określony dla zadanego kontekstu, tzn. istnieją stałe wartości parametrów rozkładu, dla których próby z rozkładu są powtarzalne. Jeśli więc rozpatrzymy dwie próby losowe, to korzystając z odpowiedniej miary niepodobieństwa rozkładów można stwierdzić, czy pochodzą one z jednego, czy z dwóch różnych rozkładów. Wniosek ten opiera się na założeniu o powtarzalności prób w modelowaniu częstościowym.

[2]Stosując wzór Bayesa dla rozkładu łącznego mamy $p(\mathbf{u}, y|c_m) = p(y|\mathbf{u}, c_m) \cdot p(\mathbf{u}|c_m)$.

Zatem, uwzględniając fakty podane w poprzednim punkcie, wnioskowanie o zmianach dla ciągu obserwacji w ujęciu częstościowym można sformułować w następujący sposób:

1. Dla dwóch okien przesuwnych, dla każdego okna z osobna oszacuj rozkłady prawdopodobieństwa dotyczące wyjścia za pomocą histogramów.

2. Jeżeli różnica między rozkładami jest większa od zadanej wartości parametru wrażliwości, to zgłoś zmianę kontekstu.

Innymi słowy, jeśli rozkłady różnią się znacząco, to zaszła zmiana kontekstu. Zakładamy, że kontekst oraz moment zmiany są wielkościami deterministycznymi.

Uwaga 1. Prezentowany algorytm w ujęciu częstościowym jest zbliżony do metod przedstawionych w literaturze przedmiotu. Jednak jego nowość polega na uwzględnieniu faktów dotyczących sposobu modelowania rozkładów (rozpatrywanie wyłącznie prawdopodobieństw warunkowych, zastosowanie histogramów, stosowanie dwóch sąsiadujących okien przesuwnych) oraz użyciu miar niepodobieństwa wcześniej nierozpatrywanych w odniesieniu do wykrywania zmian (miara Bhattacharyya, miara Lina-Wonga).

Uwaga 2. Ze względu na przejrzystość zapisu dalej pomijamy warunkowanie kontekstem, tzn. zamiast $p(y|\mathbf{u}, c_m)$ używać będziemy $p(y|\mathbf{u})$.

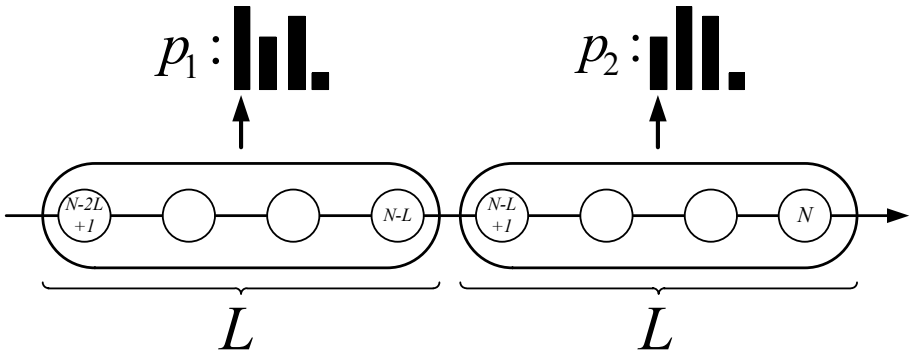
3.3.1 Technika szacowania prawdopodobieństw

Technika estymacji prawdopodobieństw opiera się na dwóch oknach przesuwnych [141]. Dla każdego okna z osobna szacowany jest rozkład prawdopodobieństwa z wykorzystaniem histogramów. Zakładając, że oba okna są tej samej długości L, postępowanie jest następujące:

1. Oszacuj prawdopodobieństwo p_1 za pomocą histogramu w oparciu o obserwacje zawarte w pierwszym oknie $\mathcal{D}_{N-L}^{L}$.

2. Oszacuj prawdopodobieństwo p_2 za pomocą histogramu w oparciu o obserwacje zawarte w drugim oknie $\mathcal{D}_{N}^{L}$.

Przykładowe okna przedstawione są na rysunku 3.1, gdzie pojedyncze kółko reprezentuje jedną obserwację, natomiast elipsoidy oznaczają dwa sąsiadujące okna przesuwne, każde o długości L.



Rysunek 3.1: Dwa sąsiadujące okna przesuwne.

Warto zaznaczyć, że im dłuższe okno L, tym większe opóźnienie wykrycia zmiany kontekstu. Faktyczna zmiana znajduje się mniej więcej w momencie $(N - L + 1)$

3.3.2 Miary niepodobieństwa

W celu porównania rozkładów $p_1(y|\mathbf{u})$ i $p_2(y|\mathbf{u})$ najprościej jest stosować miarę odległości zadaną na przestrzeni rozkładów prawdopodobieństwa, np. metrykę definiowaną przez normę l_1. W teorii decyzji miara związana z normą l_1 określana jest jako *bayesowskie prawdopodobieństwo popełenienia błędu decyzji*, które jest definiowane w następujący sposób [12, 13, 39, 76, 111]:

$$P_e(p_1, p_2) = \sum_{y \in \mathscr{Y}} \min \Big\{ \pi(p_1)\ p_1(y|\mathbf{u}), \big(1 - \pi(p_1)\big)\ p_2(y|\mathbf{u}) \Big\}, \tag{3.1}$$

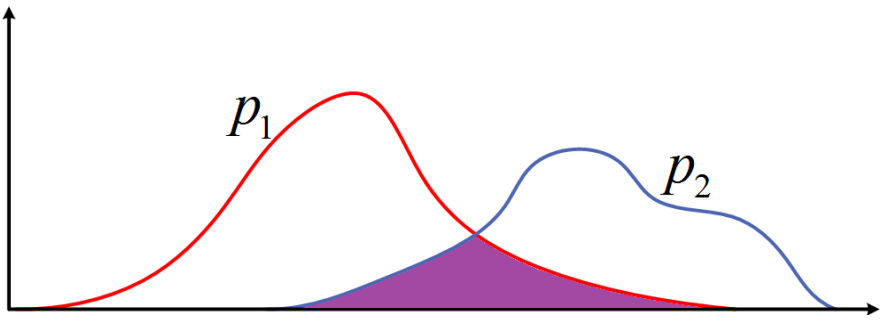
gdzie $\pi(p_1)$ oznacza prawdopodobieństwo a priori wystąpienia p_1. Jeśli p_1 i p_2 nie pokrywają się, tzn. $\sum_y p_1 p_2 = 0$, to $P_e = 0$ i wówczas zmiana kontekstu na pewno występuje. Natomiast gdy rozkłady p_1 i p_2 w pełni pokrywają się, to $P_e = 1$ i zmiana kontekstu nie występuje[3].

[3]Mylące może być, że nazwa miary P_e odwołuje się do **błędu**. Jednak z punktu widzenia wykrywania zmian wartość tej miary określa prawdopodobieństwo błędu, że dwa rozkłady mogą być „uznane za takie

Jeżeli $\pi(p_1) = 1/2$ (równe prawdopodobieństwa a priori), to prawdopodobieństwo P_e przyjmuje następującą postać [39, 76]:

$$P_e(p_1, p_2) = \frac{1}{2} \sum_{y \in \mathscr{Y}} \min \Big\{ p_1(y|\mathbf{u}), p_2(y|\mathbf{u}) \Big\}$$
$$= \frac{1}{2} - \frac{1}{4} \sum_{y \in \mathscr{Y}} |p_1(y|\mathbf{u}) - p_2(y|\mathbf{u})|. \tag{3.2}$$

Założenie jednakowych wartości rozkładów a priori powoduje, że prawdopodobieństwo P_e jest ściśle związane z normą l_1 [39]. Na rysunku 3.2 przedstawiono wartość prawdopodobieństwa popełnienia błędu (kolor fioletowy) dla dwóch przykładowych rozkładów prawdopodobieństwa.



Rysunek 3.2: Przykład różnicy dwóch rozkładów prawdopodobieństwa wyrażonych za pomocą prawdopodobieństwa popełnienia błędu P_e (kolor fioletowy).

Jednak stosowanie w praktyce miary P_e związane jest z dwoma trudnościami [76]. Po pierwsze, operowanie na normie l_1 wiąże się z trudnościami analitycznymi. Po drugie, w praktyce p_1 i p_2 są oszacowaniami rzeczywistych rozkładów. W związku z tym faktem stosowanie normy l_1 skutkuje znacznymi wartościami dla niewielkich różnic rozkładów. Innymi słowy, miara P_e zdefiniowana jak (3.2) zmienia się liniowo ze względu na różnice w rozkładach i dla oszacowań empirycznych może prowadzić do wskazywania momentów zmian kontekstu wynikających z drobnych błędów oszacowania rozkładów.

same". Zatem dla przypadku, gdy $P_e = 1$ mamy, że rozkłady są takie same, czyli z prawdopodobieństwem równym 1 popełniamy błąd przy ich rozróżnianiu. Innymi słowy oznacza to, że dla dwóch sąsiadujących okien przesuwnych dane są wartościami zmiennej losowej zadanymi z tego samego rozkładu prawdopodobieństwa.

Dlatego dalej rozważono miary niepodobieństwa rozkładów prawdopodobieństwa, które ograniczają z góry miarę P_e oraz jednocześnie są mniej wrażliwe na niewielkie różnice rozkładów. Takimi miarami są miara Bhattacharyya, ϱ_B, oraz Lina-Wonga, ϱ_{LW}. Dodatkowo zaproponowano modyfikację miary ϱ_{LW}. Ponadto, ze względów teoretycznych, przedstawiono miarę Kołmogorowa ϱ_K, oraz zastosowano miarę Kullbacka-Leiblera, ϱ_{KL}.

Miara Kołmogorowa

Miarę Kołmogorowa definiuje się w następujący sposób [41, 76, 139]:

$$\varrho_K(p_1, p_2) = \sum_{y \in \mathscr{Y}} |p_1(y|\mathbf{u}) - p_2(y|\mathbf{u})|, \tag{3.3}$$

gdzie $0 \leqslant \varrho_K \leqslant 2$, p_1 i p_2 to prawdopodobieństwa.

Zauważmy, że miara Kołmogorowa jest metryką Manhattan, czyli definiuje miarę P_e [76]

$$P_e(p_1, p_2) = \frac{1}{2}\Big(1 - \frac{1}{2}\varrho_K(p_1, p_2)\Big). \tag{3.4}$$

Fakt ten wykorzystamy w celu pokazania własności miar niepodobieństwa rozkładów prawdopodobieństwa.

Miara Bhattacharyya

Współczynnik Bhattacharyya definiuje się w następujący sposób [41, 73, 139]:

$$B(p_1, p_2) = \sum_{y \in \mathscr{Y}} \sqrt{p_1(y|\mathbf{u}) \cdot p_2(y|\mathbf{u})}, \tag{3.5}$$

gdzie $B \in [0, 1]$. Wówczas miarę Bhattacharyya określa się w następujący sposób[4] [73, 139], tzn.

$$\varrho_B(p_1, p_2) = -\ln\Big(B(p_1, p_2)\Big), \tag{3.6}$$

gdzie $0 \leqslant \varrho_B(p_1, p_2) < \infty$. Warto również pamiętać, że wyznacznik Bhattacharyya jest wyrażony przez miarę Bhattacharyya w następujący sposób

$$B(p_1, p_2) = \exp\{-\varrho_B(p_1, p_2)\}. \tag{3.7}$$

[4]Miara ta nie jest metryką, ponieważ nie spełnia nierówności trójkąta [73]. Można podać miarę, która jest metryką, np. $\sqrt{1 - B}$ [73].

Współczynnik Bhattacharyya ogranicza od dołu miarę Kołmogorowa (przy założeniu jednakowych prawdopodobieństw hipotez) [73, 139]

$$\frac{1}{2}\varrho_K \leqslant \sqrt{1-B^2} \tag{3.8}$$

i stąd

$$B \leqslant \sqrt{1-\frac{1}{4}\varrho_K^2}. \tag{3.9}$$

oraz ogranicza od dołu i od góry miarę P_e [73, 76]:

$$\left(\frac{1}{2}B(p_1,p_2)\right)^2 \leqslant P_e(p_1,p_2) \leqslant \frac{1}{2}B(p_1,p_2). \tag{3.10}$$

Korzystając z (3.7) można wyrazić (3.10) przy pomocy ϱ_B.

Miara Kullbacka-Leiblera

Miarę Kullbacka-Leiblera definiuje się w następujący sposób [41, 94, 139]:

$$\varrho_{KL}(p_1,p_2) = \sum_{y\in\mathscr{Y}} p_1(y|\mathbf{u}) \log \frac{p_1(y|\mathbf{u})}{p_2(y|\mathbf{u})}, \tag{3.11}$$

gdzie logarytm jest zazwyczaj o bazie 2.

Miara ta jest niezdefiniowana wtedy, gdy $p_1(y|\mathbf{u}) = 0$ i $p_2(y|\mathbf{u}) = 0$, oraz jest nieujemna, addytywna i niesymetryczna. Czasem definiuje się symetryczną miarę:

$$\varrho_J(p_1,p_2) = \varrho_{KL}(p_1,p_2) + \varrho_{KL}(p_2,p_1). \tag{3.12}$$

Miara ϱ_{KL} zwana jest też entropią względną i określa ile średnio należy użyć dodatkowych bitów do zakodowania próbki z p_1 korzystając z kodu w oparciu o p_2.

W [111] pokazano, że niemożliwe jest określenie górnego oszacowania na P_e przy użyciu miary Kullbacka-Leiblera; można jedynie podać ograniczenie od dołu [76, 94].

Miara Lina-Wonga

Miarę Lina-Wonga definiuje się w następujący sposób [41, 94]:

$$\varrho_{LW}(p_1,p_2) = \sum_{y\in\mathscr{Y}} p_1(y|\mathbf{u}) \log \frac{p_1(y|\mathbf{u})}{\frac{1}{2}p_1(y|\mathbf{u}) + \frac{1}{2}p_2(y|\mathbf{u})}, \tag{3.13}$$

gdzie logarytm jest zazwyczaj o bazie 2.

Warto zauważyć, że $\varrho_{LW}(p_1, p_2) = \varrho_{KL}(p_1, \frac{1}{2}p_1 + \frac{1}{2}p_2)$, oraz $0 \leqslant \varrho_{LW} \leqslant 1$ i jest dobrze zdefiniowana dla dowolnych wartości $p_1(y|\mathbf{u})$ i $p_2(y|\mathbf{u})$ [94], ale jest niesymetryczna[5].

Miara Lina-Wonga posiada dwie istotne własności [41, 94]:

$$\varrho_{LW}(p_1, p_2) \leqslant \frac{1}{2}\varrho_{KL}(p_1, p_2) \tag{3.14}$$

oraz

$$\varrho_{LW}(p_1, p_2) \leqslant \frac{1}{2}\varrho_{K}(p_1, p_2). \tag{3.15}$$

Można pokazać następującą własność dotyczącą ograniczenia od góry P_e przez miarę Lina-Wonga:

Lemat 3.1. *Niech $\pi(p_1) = 1/2$, wówczas zachodzi następująca nierówność*

$$P_e(p_1, p_2) \leqslant \frac{1}{2}\Big(1 - \varrho_{LW}(p_1, p_2)\Big).$$

Dowód. Z własności (3.15) mamy

$$\varrho_{LW}(p_1, p_2) \leqslant \frac{1}{2}\varrho_{K}(p_1, p_2).$$

Przekształcając otrzymujemy

$$\frac{1}{2}\Big(1 - \varrho_{LW}(p_1, p_2)\Big) \geqslant \frac{1}{2}\Big(1 - \frac{1}{2}\varrho_{K}(p_1, p_2)\Big).$$

Z (3.4) widać, że prawa strona nierówności jest definicją $P_e(p_1, p_2)$, co kończy dowód. □

Zmodyfikowana miara Lina-Wonga

Ze względu na szacowanie rozkładów prawdopodobieństwa warto również zastanowić się nad taką miarą, która jest mniej wrażliwa na drobne zmiany. W tym celu proponowana jest zmodyfikowana miara Lina-Wonga. Ponieważ $\varrho_{LW} \in [0, 1]$, więc podniesienie wartości miary Lina-Wonga do kwadratu spowoduje jej *spłaszczenie* dla wartości bliskich zeru. Zabieg ten prowadzi do mniejszej wrażliwości na drobne zmiany i wzmocnienie istotnych.

[5] Symetryczność osiąga się w analogiczny sposób jak we wzorze (3.12).

Zmodyfikowaną miarę Lina-Wonga definiujemy w następujący sposób

$$\varrho_{LW2}(p_1, p_2) = \Big(\varrho_{LW}(p_1, p_2)\Big)^2. \tag{3.16}$$

Warto zauważyć, że wzięcie kwadratu z miary $\mathcal{D}_{LW}$ zachowuje jej własności z kwadratem, tzn.

$$\varrho_{LW2}(p_1, p_2) \leqslant \frac{1}{4}\Big(\varrho_{KL}(p_1, p_2)\Big)^2 \tag{3.17}$$

oraz

$$\varrho_{LW2}(p_1, p_2) \leqslant \frac{1}{4}\Big(\varrho_{K}(p_1, p_2)\Big)^2. \tag{3.18}$$

Można pokazać następującą własność dotyczącą ograniczenia od góry P_e przez zmodyfikowaną miarę Lina-Wonga:

Lemat 3.2. *Niech $\pi(p_1) = 1/2$, wówczas zachodzi następująca nierówność*

$$P_e(p_1, p_2) \leqslant \frac{1}{2}\sqrt{1 - \varrho_{LW2}(p_1, p_2)}.$$

Dowód. Z własności (3.18) mamy

$$\varrho_{LW2}(p_1, p_2) \leqslant \frac{1}{4}\Big(\varrho_{K}(p_1, p_2)\Big)^2.$$

Przekształcając otrzymujemy

$$\sqrt{1 - \varrho_{LW2}(p_1, p_2)} \geqslant \sqrt{1 - \frac{1}{4}\Big(\varrho_{K}(p_1, p_2)\Big)^2}.$$

Korzystając z (3.9) widać, że lewa strona ogranicza z góry współczynnik Bhattacharyya, czyli

$$\sqrt{1 - \varrho_{LW2}(p_1, p_2)} \geqslant B(p_1, p_2).$$

Zatem ostatecznie otrzymujemy górne ograniczenie na P_e z nierówności (3.10), co kończy dowód. □

3.3.3 Algorytm wykrywania zmian w podejściu częstościowym

Po określeniu techniki szacowania prawdopodobieństwa oraz opisaniu miar, algorytm wykrywania zmian kontekstu jest określony jak w procedurze 3.3.1.

Algorytm 3.3.1. Algorytm wykrywania zmian kontekstu w podejściu częstościowym.

Wejście: (i) ciąg obserwacji $\mathcal{D}$, (ii) L, (iii) σ, (iv) $\varrho(\cdot,\cdot)$; (v) $\tau = \emptyset$, (vi) $N := 0$.

Wyjście: Momenty zmian kontekstu τ.

Krok 1: Ustaw $N := N + 1$. Jeśli $N > \mathrm{card}\{\mathcal{D}\}$, to STOP.

Krok 2: Oszacuj prawdopodobieństwo p_1 na podstawie danych $\mathcal{D}_{N-L}^{L}$, oraz prawdopodobieństwo p_2 na podstawie danych $\mathcal{D}_{N}^{L}$.

Krok 3: Wyznacz wartość miary $\varrho(p_1, p_2)$.

Krok 4: Sprawdź, czy $\varrho(p_1, p_2) \geqslant \sigma$. Jeśli tak, to zgłoś zmianę, poszerz zbiór momentów zmian $\tau = \tau \cup \{N - L + 1\}$. Idź do kroku 1.

Uwaga. W kroku 2, w przypadku $N < 2L$, należy brać wszystkie obserwacje od początku i ustawić oba okna na tych samych obserwacjach. Podobnie w sytuacji, gdy nastąpi wykrycie zmiany. Wówczas należy ustawić okna przesuwne w momencie wykrytej zmiany i stopniowo zwiększać je aż do uzyskania założonej długości okien. Następnie tylko okno z obserwacjami do oszacowania p_2 jest przesuwane tak długo, aż okna będą rozłączne. Dopiero wtedy oba okna są przesuwane razem.

3.4 Podejście bayesowskie[6]

Ogólnie mówiąc, w modelowaniu bayesowskim zakłada się, że wszystkie modelowane wielkości są zmiennymi losowymi [12, 13, 53]. Zatem zarówno liczba zmian, momenty zmian oraz konteksty są zmiennymi losowymi.

W przypadku z ciągiem obserwacji i zmiennym kontekstem przyjmuje się, że ciąg obserwacji $\mathcal{D}$ można podzielić na M kontekstów. Dla każdego $\mathcal{D}_m$ zakłada się, że obserwacje są niezależnie i o jednakowym rozkładzie (iid), co oznacza, że są próbą z rozkładu $p(y|\mathbf{u}, \boldsymbol{\theta}_m)$, gdzie $\boldsymbol{\theta}_m$ oznacza wektor parametrów dla m-tego kontekstu.

[6]W niniejszym rozdziale przyjmujemy następującą konwencję: argument rozkładu rozróżnia rozkłady, indeks przy zmiennej lub parametrze rozróżnia zmienne lub parametry.

Proponowane podejście do wykrywanie zmian kontekstu jest więc następujące. Przyjmujemy okno przesuwne o długości L i wówczas, korzystając z założeń o iid, możliwe są dwa przypadki. W pierwszym dane zawarte w oknie należą w całości do jednego przedziału, natomiast w drugim obserwacje uwzględniają dwa przedziały, czyli innymi słowy – zmianę kontekstu. Milcząco zakładamy, że zmiany zachodzą na tyle rzadko, iż obserwacje z okna nie uwzględniają więcej niż jednej zmiany kontekstu. Następnie oba modele są porównywane przy pomocy współczynnika Bayesa (ang. *Bayes factor*) [53, 74], który określa stosunek wiarygodności modeli (ang. *model evidence* lub *marginal likelihood*). Współczynnik Bayesa spełnia rolę miary niepodobieństwa w podejściu częstościowym. Gdy wartość współczynnika Bayesa modelu ze zmianą w stosunku do modelu bez zmiany wynosi więcej niż zadana wartość parametru wrażliwości, to zgłaszana jest zmiana kontekstu.

3.4.1 Modelowanie bayesowskie zmian kontekstu

Przed przedstawieniem szczegółów modelowania, dokonajmy zmiany sposobu zapisu zmiennej wyjściowej y, która przyjmuje Y wartości. W literaturze często stosowany jest schemat 1-na-Y, który zmienną dyskretną reprezentuje za pomocą Y-wymiarowego wektora $\mathbf{y}$, w którym tylko jeden element na k-tej pozycji, y^k, przyjmuje wartość 1, natomiast wszystkie pozostałe 0. Wówczas rozkład prawdopodobieństwa zmiennej $\mathbf{y}$ można przedstawić w następującej formie

$$p(\mathbf{y}|\mathbf{u}, \boldsymbol{\theta}) = \prod_{k=1}^{Y} \left\{\theta_k(\mathbf{u})\right\}^{y^k}, \tag{3.19}$$

gdzie $\boldsymbol{\theta} = \left(\theta_1(\mathbf{u}) \dots \theta_Y(\mathbf{u})\right)^{\mathrm{T}}$ oraz $\theta_k(\mathbf{u})$ spełnia ograniczenia: $\theta_k(\mathbf{u}) \geqslant 0$ i $\sum_k \theta_k(\mathbf{u}) = 1$. Czyli $\theta_k(\mathbf{u})$ oznacza prawdopodobieństwo $p(y^k = 1|\mathbf{u})$. Dalej, dla przejrzystości, pisać będziemy $\theta_k(\mathbf{u}) = \theta_k$.

W proponowanym podejściu do wykrywania zmian z zastosowaniem modelowania bayesowskiego zakładamy jedno okno przesuwne o długości L zawierające obserwacje od momentu $n - L$ do n, gdzie n – bieżący moment, $\mathcal{D}_n^L$. Funkcja wiarygodności przyjmuje wówczas postać

$$p(\mathcal{D}_n^L|\boldsymbol{\theta}) = \prod_{k=1}^{Y} \theta_k^{j_k} \tag{3.20}$$

gdzie j_k oznacza liczbę wystąpień y^k wśród obserwacji zawartych w oknie przesuwnym.

Kluczowym elementem wnioskowania jest określenie postaci rozkładów dla przypadku bez zmiany kontekstu i ze zmianą. Modele te są następujące:

1. Jeśli obserwacje zawarte w oknie przesuwnym należą do jednego przedziału (brak zmiany), to funkcja wiarygodności dla modelu $\mathcal{M}_0$ przyjmuje postać

$$p(\mathcal{D}_n^L|\mathcal{M}_0, \boldsymbol{\theta}_0) = p(\mathcal{D}_n^L|\boldsymbol{\theta}_0), \tag{3.21}$$

   gdzie $\boldsymbol{\theta}_0$ – parametry modelu $\mathcal{M}_0$.

2. Jeśli obserwacje zawarte w oknie przesuwnym należą do dwóch przedziałów (występuje zmiana w momencie t i $n - L < t \leqslant n$), to funkcja wiarygodności dla modelu $\mathcal{M}_1$ jest następująca (korzystając z założenia o iid obserwacji dla każdego c_m)

$$p(\mathcal{D}_n^L|\mathcal{M}_1, \boldsymbol{\theta}_1, t, n) = p(\mathcal{D}_t^{L-n+t}|\boldsymbol{\theta}_1^1)\; p(\mathcal{D}_n^{n-t}|\boldsymbol{\theta}_1^2), \tag{3.22}$$

   gdzie $\boldsymbol{\theta}_1 = (\boldsymbol{\theta}_1^1\ \boldsymbol{\theta}_1^2)^{\mathrm{T}}$ – parametry modelu $\mathcal{M}_1$, oraz $\boldsymbol{\theta}_1^1$ odpowiadają parametrom przed zmianą kontekstu, zaś $\boldsymbol{\theta}_1^2$ – po zmianie kontekstu. Zakładamy, że $\boldsymbol{\theta}_1^1, \boldsymbol{\theta}_1^2, t$ są niezależne.

Wówczas wiarygodność modelu $\mathcal{M}_0$ można policzyć korzystając z wyrażenia

$$p(\mathcal{D}_n^L|\mathcal{M}_0) = \int p(\mathcal{D}_n^L|\mathcal{M}_0, \boldsymbol{\theta}_0)\; p(\boldsymbol{\theta}_0|\mathcal{M}_0)\; \mathrm{d}\boldsymbol{\theta}_0, \tag{3.23}$$

gdzie $p(\boldsymbol{\theta}_0|\mathcal{M}_0)$ – rozkład *a priori* parametrów, natomiast wiarygodność modelu $\mathcal{M}_1$ z następującego wzoru (korzystając z niezależności zmiennych $\boldsymbol{\theta}_1^1, \boldsymbol{\theta}_1^2, t$)

$$p(\mathcal{D}_n^L|\mathcal{M}_1) = \iint p(\mathcal{D}_n^L|\mathcal{M}_1, \boldsymbol{\theta}_1, t)\; p(\boldsymbol{\theta}_1^1|\mathcal{M}_1)\; p(\boldsymbol{\theta}_1^2|\mathcal{M}_1)\; p(t|\mathcal{M}_1)\; \mathrm{d}\boldsymbol{\vartheta}_1\; \mathrm{d}t, \tag{3.24}$$

gdzie $p(\boldsymbol{\theta}_1^1|\mathcal{M}_1)$, $p(\boldsymbol{\theta}_1^2|\mathcal{M}_1)$, $p(t|\mathcal{M}_1)$ – rozkłady *a priori* na parametry.

W celu wyznaczenia wiarygodności modeli należy określić rozkłady *a priori* parametrów, a następnie dokonać całkowania względem parametrów. W rozpatrywanym przypadku, dla funkcji wiarygodności o postaci (3.20), jest to zadanie bardzo trudne do analitycznego wyznaczenia. Problemem dotyczącym sposobu wyznaczenia tych wielkości zajmiemy się w następnym punkcie.

Zakładając na chwilę, że znamy wiarygodności modeli, oraz że rozkłady *a priori* modeli są sobie równe, $p(\mathcal{M}_0) = p(\mathcal{M}_1)$, w celu porównania modeli należy policzyć współczynnik Bayesa [53, 74]:

$$B_{10} = \frac{p(\mathcal{D}_n^L|\mathcal{M}_1)}{p(\mathcal{D}_n^L|\mathcal{M}_0)}. \tag{3.25}$$

Stosunek ten określa ilościowe wsparcie dla modelu $\mathcal{M}_1$, tzn. im wartość większa od 1, tym silniejszy dowód modelu, natomiast dla wartości poniżej 1 preferowany jest wybór modelu $\mathcal{M}_0$. Jeffreys podał interpretację, wg której określa się siłę wsparcia dla modelu (patrz tabela 3.1). Dobór parametru wrażliwości można dokonać w oparciu o tę interpretację.

B_{10}	$\ln(B_{10})$	Wsparcie dla modelu $\mathcal{M}_1$
$1-3$	$0-1$	Słabe
$3-10$	$1-2.3$	Pozytywne
$10-100$	$2.3-4.6$	Silne
>100	>4.6	Bardzo silne

Tabela 3.1: Interpretacja współczynnika Bayesa wg Jeffreysa [74].

Czasem wygodniej jest operować na logarytmie ze współczynnika Bayesa, tzn.

$$\ln(B_{10}) = \ln p(\mathcal{D}_n^L|\mathcal{M}_1) - \ln p(\mathcal{D}_n^L|\mathcal{M}_0). \tag{3.26}$$

Podsumowując, aby wykryć zmianę kontekstu należy wyznaczyć współczynnik Bayesa B_{10} dla obserwacji zawartych w oknie przesuwnym, a następnie sprawdzić, czy jest on większy od zadanej wartości σ, np. 2. Jeśli tak, to jest to równoznaczne z pozytywnym dowodem modelu $\mathcal{M}_1$, a co za tym idzie – zaistnieniu sytuacji, że obserwacje w oknie przesuwnym należą do dwóch kontekstów. W przeciwnym przypadku, gdy współczynnik Bayesa jest mniejszy od σ, przyjmujemy, że zmiana nie zaszła.

3.4.2 Aproksymacja wiarygodności modelu

Jak wspomniano wcześniej, wyznaczenie wiarygodności modeli jest zadaniem bardzo trudnym, czasem nawet niemożliwym analitycznie. W celu policzenia wiarygodności modeli $\mathcal{M}_0$ oraz $\mathcal{M}_1$ przyjmujemy, że

- nie ma żadnych przesłanek, aby wyrazić silniejsze przekonania odnośnie wybranych wartości parametrów $\boldsymbol{\theta}_0$ modelu $\mathcal{M}_0$, więc zakładamy nieinformacyjny rozkład *a priori* (rozkład jednostajny);
- nie ma żadnych przesłanek, aby wyrazić silniejsze przekonania odnośnie wybranych wartości parametrów $\boldsymbol{\theta}_1$ modelu $\mathcal{M}_1$, więc zakładamy nieinformacyjny rozkład *a priori* (rozkład jednostajny);
- zmiana zachodzi w środku okna przesuwnego, tj. w momencie $n - \frac{1}{2}L$, zatem przyjmujemy rozkład *a priori* dla t jako deltę Diraca w punkcie $n - \frac{1}{2}L$.

Dla takich założeń wiarygodność modelu można aproksymować za pomocą kryterium Schwarza (ang. *Schwarz Criterion*, lub *Bayesian Information Criterion*, BIC) [135]

$$\ln p(\mathcal{D}_n^L|\mathcal{M}) \approx \ln p(\mathcal{D}_n^L|\hat{\boldsymbol{\theta}}) - \frac{Y}{2}\ln L, \tag{3.27}$$

gdzie $\hat{\boldsymbol{\theta}}$ jest estymatorem maksymalnej wiarygodności.

Zatem stosując (3.27) do (3.23) oraz (3.24) otrzymujemy, odpowiednio

$$\ln p(\mathcal{D}_n^L|\mathcal{M}_0) \approx \sum_{k=1}^{Y} j_k \ln \hat{\theta}_{0,k} - \frac{Y}{2}\ln L, \tag{3.28}$$

oraz

$$\ln p(\mathcal{D}_n^L|\mathcal{M}_0) \approx \sum_{k=1}^{Y} (j_k^1 \ln \hat{\theta}_{1,k}^1 + j_k^2 \ln \hat{\theta}_{1,k}^2) - Y \ln L, \tag{3.29}$$

gdzie j_k^1 określa liczbę wystąpień y^k w pierwszej połowie okna przesuwnego, natomiast j_k^2 – w drugiej połowie okna przesuwnego.

Ostatecznie, korzystając z (3.28) i (3.29), można wyznaczyć aproksymowaną wartość współczynnika Bayesa (3.26), tj.

$$\ln B_{10} \approx \sum_{k=1}^{Y} (j_k^1 \ln \hat{\theta}_{1,k}^1 + j_k^2 \ln \hat{\theta}_{1,k}^2) - \sum_{k=1}^{Y} j_k \ln \hat{\theta}_{0,k} - \frac{Y}{2}\ln L. \tag{3.30}$$

3.4.3 Algorytm wykrywania zmian w podejściu bayesowskim

Po określeniu sposobu modelowania zmian kontekstu oraz metody, w jaki należy wyznaczyć aproksymację wiarygodności modeli, a co za tym idzie – aproksymację współczynnika Bayesa, algorytm wykrywania zmian kontekstu jest przedstawiony jak w procedurze 3.3.1.

Algorytm 3.4.1. Algorytm wykrywania zmian kontekstu w podejściu bayesowskim.

Wejście: (i) ciąg obserwacji $\mathcal{D}$, (ii) L, (iii) σ, (iv) $\tau = \emptyset$, (v) $N := 0$, (vi) postaci modeli $\mathcal{M}_0$ i $\mathcal{M}_1$.

Wyjście: Momenty zmian kontekstu τ.

Krok 1: Ustaw $N := N + 1$. Jeśli $N > \mathrm{card}\{\mathcal{D}\}$, to STOP.

Krok 2: Na podstawie danych $\mathcal{D}_N^L$ wyznacz wiarygodność modelu $\mathcal{M}_0$ korzystając z (3.28) oraz wiarygodność modelu $\mathcal{M}_1$ korzystając z (3.29).

Krok 3: Wyznacz wartość logarytmu ze współczynnika Bayesa, $\ln B_{10}$, na podstawie wyrażenia (3.30).

Krok 4: Sprawdź, czy $\ln B_{10} \geqslant \sigma$. Jeśli tak, to zgłoś zmianę, poszerz zbiór momentów zmian, $\tau = \tau \cup \{N - L/2\}$, $N := N + 1$, idź do kroku 1. W przeciwnym razie $N := N + 1$, idź do Kroku 1.

Uwaga 1. W kroku 2, w przypadku $N < L$, należy brać wszystkie obserwacje od początku. W sytuacji, gdy nastąpi wykrycie zmiany, należy ustawić okno na najnowszej obserwacji i stopniowo powiększać aż do rozmiaru L. Zmiana nie jest wówczas zgłaszana.

Uwaga 2. Ze względu na założenia, rzeczywista zmiana znajduje się w połowie okna, czyli w momencie $N - L/2$.

3.5 Uwagi

- Ekstrakcja wiedzy z wykrywaniem zmian kontekstu (tymczasowe uczenie wsadowe) jest podejściem retrospektywnym, ponieważ wiedza jest formułowana dopiero po wykryciu momentu zmiany kontekstu. Jednak prezentowane algorytmy są procedurami przyrostowymi z wykorzystaniem okien przesuwnych. Określanie momentów zmian kontekstów odbywa się na bieżąco wraz z pojawianiem się nowych obserwacji.

- Zarówno w podejściu częstościowym, jak i bayesowskim, wykrywanie zmian kontekstów odbywa się zawsze z opóźnieniem w odniesieniu do momentu pojawienia się najnowszej obserwacji.

- Konstrukcja obu algorytmów wykrywania zmian kontekstu jest zbliżona, jednak założenia i sposób wnioskowania w obu podejściach są odmienne.

- W podejściu częstościowym kluczowymi wielkościami są długość okna L oraz wartość współczynnika wrażliwości σ, którego wartość należy dopasować ze względu na stosowaną miarę oraz rozpatrywany problem. Jest to główna wada tego podejścia. Zaletą jest prostota i klarowność postępowania.

- W podejściu bayesowskim kluczowymi wielkościami są długość okna L oraz wiarygodność modeli. W celu ustalenia wartości σ można posłużyć się interpretacją Jeffreysa (patrz tabela 3.1). Główną trudnością w stosowaniu tego podejścia jest policzenie wiarygodności modeli. Natomiast zaletą są szerokie możliwości modelowania obiektu oraz uwzględnienie wiedzy apriorycznej.

- Ze względu na wykorzystanie wykrywania zmian kontekstu dla tymczasowego uczenia wsadowego, w pracy rozpatrzono wyłącznie przypadek dla zmiennej dyskretnej. Tym niemniej prezentowane podejścia można bez problemu stosować dla zmiennych ciągłych. Wówczas w modelowaniu częstościowym należy posłużyć się estymatorem maksymalnej wiarygodności do szacowania wartości parametrów rozkładów. Natomiast w modelowaniu bayesowskim – wybrać odpowiednie postaci rozkładów aprioorycznych, np. rozkłady sprzężone do rozkładów aposteriorycznych, oraz rozkłady dla modeli $\mathcal{M}_0, \mathcal{M}_1$.

Rozdział 4

Ekstrakcja wiedzy z oknem przesuwnym

Rozdział zawiera oryginalne rezultaty pracy, tzn. dwa algorytmy ekstrakcji wiedzy regułowej z oknem przesuwnym.

4.1 Wprowadzenie

Podejście prezentowane w poprzednim rozdziale, tzn. podejście z tymczasowym uczeniem wsadowym, jest przydatne, gdy wiedza jest ekstrahowana w sposób retrospektywny. Oznacza to, że reguły są indukowane dla każdego kontekstu z osobna. Jednakże w przypadku, gdy wiedza potrzebna jest do bieżącego podejmowania decyzji, to należy korzystać z podejścia prospektywnego, czyli uczenia z uaktualnianiem.

Kolejnym powodem, dla którego stosuje się podejście z uaktualnianiem, jest fakt, że problem ekstracji wiedzy regułowej jest problemem NP-zupełnym [2]. Zatem dla indukcji reguł dla obserwacji pochodzących z jednego kontekstu czas potrzebny na wyekstrahowanie wiedzy może być zbyt duży.

Zatem w niniejszym rozdziale zaproponowano algorytmy z oknem przesuwnym w celu umożliwienia podejmowania decyzji na podstawie wiedzy na bieżąco oraz aby zniwelować czas uczenia. Metody te nazywamy:

1. Algorytm AQ-P1 – propozycja pierwsza algorytmu z oknem przesuwnym, który do indukcji reguł wykorzystuje algorytm AQ.

2. Algorytm AQ-P2 – propozycja druga algorytmu z oknem przesuwnym oraz zapominaniem wybiórczym na modelu, który do indukcji reguł wykorzystuje algorytm AQ.

W obu algorytmach zastosowano schemat, który składa się z dwóch podstawowych kroków [25, 26]:

1. Walidacja wiedzy (ang. *knowledge validation*).

2. Uaktualnianie wiedzy (ang. *knowledge updating*).

Krok walidacji polega na sprawdzeniu, czy w świetle nowych obserwacji wiedza o obiekcie nadal jest przydatna wg zadanej funkcji oceniającej przydatność. Następnie wiedza lub jej część jest usuwana. Natomiast uaktualnianie wiedzy następuje zawsze po pojawieniu się nowych obserwacji i polega na ekstrahowaniu nowej wiedzy o obiekcie i połączeniu jej z poprzednio otrzymaną wiedzą.

Uwaga. Ze względu na fakt, iż algorytm AQ odgrywa kluczową rolę w działaniu dwóch proponowanych algorytmów, został on dokładniej omówiony w Dodatku.

4.2 Algorytm AQ-P1

Pierwsza propozycja algorytmu dla ekstrakcji wiedzy regułowej polega na tym, aby ekstrahować wiedzę na podstawie obserwacji zawartych w oknie przesuwnym, a następnie sprawdzać, czy na podstawie otrzymanej wiedzy podejmowane są poprawne decyzje (walidacja). Dopiero wtedy, gdy zostanie popełniona odpowiednia liczba błędów, wiedza jest usuwana i reguły są indukowane na podstawie ostatnich L obserwacji (uaktualnianie).

Kroki algorytmu przedstawiono w procedurze 4.2.1.

Algorytm 4.2.1. Algorytm AQ-P1.

Wejście: (i) ciąg obserwacji $\mathcal{D}$, (ii) kryterium jakości 1.14, (iii) L – długość okna przesuwnego, (iv) η – liczba ostatnio sprawdzanych klasyfikacji, (v) $AQ(\cdot,\cdot)$ – algorytm indukcji reguł AQ, (vi) $\Phi_N := \emptyset$, (vii) $N := 0$.

Wyjście: Zestaw reguł Φ_N.

Krok 1: Ustaw $N := N + 1$. Jeśli $N > \mathrm{card}\{\mathcal{D}\}$, to STOP.

Krok 2: (`Walidacja`) Sprawdź, czy na podstawie obserwacji $\mathcal{D}_N^\eta$ kryterium jakości rośnie. Jeśli tak, to $\Phi_N := \Phi_{N-1}$ i idź do kroku 1.

Krok 3: (`Uaktualnianie`) Ekstrahuj reguły na podstawie $\mathcal{D}_N^L$, tj.

$$\Phi_N := AQ(\mathcal{D}_N^L).$$

Idź do kroku 1.

Uwagi:

1. Dla algorytmu 4.2.1 walidacja wiedzy o obiekcie polega sprawdzeniu liczby błędów na podstawie η ostatnich obserwacji.

2. Jeżeli walidacja wiedzy wskazuje na zwiększenie liczby błędnych decyzji podejmowanych na podstawie wiedzy, jest ona ekstrahowana od początku. Dzięki takiemu postępowaniu zawsze dysponujemy aktualną wiedzą o obiekcie.

3. Ze względu na fakt, że wiedza ekstrahowana jest za każdym razem od początku, długość okna powinna być najbliższa wartości wymiaru Vapnika-Chervonenkisa dla rozpatrywanego problemu (patrz twierdzenie Vapnika-Chervonenkisa i wnioski podane w punkcie 2.2 pracy). Wówczas średni błąd podejmowania decyzji będzie najmniejszy. Takie podejście zwraca najlepsze wyniki nawet wówczas, gdy zmiana kontekstu będzie zachodziła częściej niż wymiar *VC-dim*.

4. W podanym algorytmie, w kroku 3, można korzystać z innego niż AQ algorytmu indukcji reguł, np. CN2.

5. Algorytm 4.2.1 można zmodyfikować poprzez dołączenie zapominania wybiórczego w kroku walidacji.

6. Złożoność obliczeniowa algorytmu zależy od metody wykorzystanej w kroku uaktualniania. Złożoność obliczeniową specjalizacji pojedynczej reguły w algorytmie AQ można oszacować korzystając z notacji duże „O”, czyli złożoności asymptotycznej,

wówczas $O\Big(2^{K_\Delta} D\Big(N + \log(2^{K_\Delta} D)\Big)\Big)$ [34], gdzie $K_\Delta = \max\limits_{d=1,2,\ldots,D} K_d$, D – liczba cech, N – liczba obserwacji. Zatem wygenerowanie pojedynczej reguły jest wykładnicze względem wartości atrybutów i dodatkowo mnożona przez liczbę obserwacji.

7. Przedstawiony algorytm jest algorytmem ze stałym oknem przesuwnym. Jednak można stosować dodatkowy mechanizm, który mógłby modyfikować długość okna w celu dostosowywania się do zmian kontekstu. W tym celu można stosować dodatkową metodę wykrywania zmian kontekstów, który po zgłoszeniu zmiany mógłby zmniejszać długość okno do zadanej wartości $\hat{L}$, a następnie w każdym kroku zwiększać ją aż do osiągnięcia wartości L.

4.3 Algorytm AQ-P2

Druga propozycja algorytmu ekstrakcji wiedzy w postaci regułowej opiera się na schemacie algorytmu 4.2.1 z oknem przesuwnym, jednak zmodyfikowany jest krok walidacji wiedzy. Modyfikacja polega na wprowadzeniu mechanizmu zapominania wybiórczego. Ogólna idea algorytmu polega na indukcji reguł na podstawie obserwacji z okna przesuwnego w każdym momencie. W kroku walidacji część reguł jest usuwana, natomiast w kroku uaktualniania wiedza z usuniętymi regułami jest łączona z wiedzą wyekstrahowaną z okna przesuwnego.

Zapominanie wybiórcze w proponowanym algorytmie opiera się na eliminowaniu reguł z modelu, dla których wartość zadanej *funkcji oceniającej* regułę jest mniejsza niż określona wielkość. W literaturze można znaleźć wiele różnych funkcji oceniających reguły, m.in. *pokrycie* i *dokładność* [117]. Jednak dalej skorzystamy z następującej funkcji oceniającej regułę na podstawie obserwacji z okna przesuwnego o długości L [51]:

$$f(\phi, \mathcal{D}_n^L) = \frac{N_\phi^l + (\bar{N} - \bar{N}_\phi)}{L} \tag{4.1}$$

gdzie n – aktualny moment, ϕ – reguła o postaci $\phi_{in} \Rightarrow \alpha_l^{out}$, l – numer klasy, N_ϕ^l – liczba obserwacji z okna przesuwnego pokrytych przez regułę ϕ z l-tej klasy, $\bar{N}_\phi$ – liczba obserwacji z okna przesuwnego pokrytych przez regułę ϕ z pozostałych klas, $\bar{N}$ – liczba obserwacji nienależących do l-tej klasy, L – liczba rozpatrywanych obserwacji w oknie przesuwnym.

Wybór tej funkcji nie jest przypadkowy, ponieważ promuje ona nie tylko pokrycie klasy, ale również brak pokrycia obserwacji z innych klas. Podobny sposób oceny został wykorzystany w uczeniu wyrażenia niefunkcyjnego, gdzie jednocześnie istotne było pokrywanie przestrzeni pozytywnej oraz wykluczanie negatywnej [25]. Sposób, w jaki funkcja (4.1) ocenia pojedynczą regułę, jest szczególnie istotne z punktu widzenia podejścia z oknem przesuwnym, w którym zawarta jest tylko niewielka część przestrzeni wszystkich obserwacji.

Warto również zauważyć, że funkcja f przyjmuje wartości z przedziału $[0, 1]$. W przypadku, gdy reguła pokrywa wyłącznie obserwacje nienależące do l-tej klasy, to $N_\phi^l = 0$, $\bar{N}_\phi = \bar{N}$ i $f = 0$. Natomiast gdy pokrywane są wyłącznie obserwacje z l-tej klasy, to $\bar{N}_\phi = 0$, $N_\phi^l + \bar{N} = N$ oraz $f = 1$. W pozostałych przypadkach $f \in (0, 1)$.

Procedurę metody AQ-P2 przedstawiono w procedurze 4.3.1.

Algorytm 4.3.1. Algorytm AQ-P2.

Wejście: (i) ciąg obserwacji $\mathcal{D}$, (ii) funkcja oceniająca 4.1, (iii) L – długość okna przesuwnego, (iv) η – liczba obserwacji dla oceny reguły, (v) $\varepsilon \in (0, 1)$ – wartość progowa dla oceniania reguł, (vi) $N := 0$, (vii) $\Phi_N := \emptyset$, (viii) $\tilde{\Phi}_N := \emptyset$, (ix) $AQ(\cdot)$ – algorytm indukcji reguł AQ.

Wyjście: Zestaw reguł Φ_N.

Krok 1: Ustaw $N := N + 1$. Jeśli $N > \mathrm{card}\{\mathcal{D}\}$, to STOP.

Krok 2: (`Walidacja`) Z modelu Φ_{N-1} usuń wszystkie reguły, dla których wartość funkcji oceniającej jest mniejsza niż zadana wartość, tj.

$$\tilde{\Phi}_{N-1} := \Phi_{N-1} \setminus \{\phi \in \Phi_{N-1} : f(\phi, \mathcal{D}_N^\eta) < \varepsilon\}.$$

Krok 3: (`Uaktualnianie etap 1`) Ekstrahuj reguły na podstawie $\mathcal{D}_N^L$,

$$\tilde{\Phi}_N := AQ(\mathcal{D}_N^L).$$

Krok 4: (`Uaktualnianie etap 2`) Wyznacz

$$\Phi_N := \tilde{\Phi}_{N-1} \cup \tilde{\Phi}_N.$$

Idź do kroku 1.

Uwagi:

1. Walidacja wiedzy polega na usuwaniu pojedynczych reguł, dla których wartość funkcji oceniającej f jest poniżej zadanego progu ε.

2. Uaktualnianie wiedzy składa się z dwóch etapów. Najpierw ekstrahowana jest nowa wiedza na podstawie okna przesuwnego. Następnie wiedza po walidacji łączona jest z nowo wyekstrahowaną wiedzą. Warto zaznaczyć, że takie postępowanie prowadzi do reguł, które mogą częściowo wzajemnie się pokrywać. Ponadto, otrzymany zbiór reguł jest zazwyczaj bardziej liczny niż np. wiedza uzyskana w wyniku algorytmu AQ-P1. Jednak takie podejście ma na celu zwiększenie jakości wiedzy, jak również możliwość zachowania części wiedzy z przeszłości.

3. Uwagi 4 i 6 dla algorytmu AQ-P1 mogą być powtórzone dla algorytmu AQ-P2.

4. Stosowanie mechanizmu do modyfikacji długości okna przesuwnego niekoniecznie może dawać pozytywne rezultaty. Wynika to z faktu wykorzystania funkcji oceniającej, tzn. w przypadku, gdy nastąpi zmiana kontekstu, to i tak część reguł zostanie wyeliminowana z modelu z powodu niskiej wartości funkcji (4.1).

5. W celu otrzymania wiedzy, która w sensie średnim zwraca mniej błędów, można wykorzystać wniosek dot. wymiaru *VC-dim*. Jednak sposób ekstrahowania wiedzy w kroku uaktualniania algorytmu AQ-P2, tzn. dołączanie nowych reguł, prowadzi do tego, że okno przesuwne może być mniejsze niż w przypadku algorytmu AQ-P1, ponieważ część wiedzy pozostaje ze starszego modelu. Innymi słowy, przestrzeń obserwacji jest częściowo pokrywana z kroku na krok, natomiast pokrycia o niskiej wartości funkcji oceny są usuwane i w ich miejsce pojawiają nowe, *najlepsze* pokrycia w danym momencie.

6. W porównaniu z algorytmem AQ-P1, ekstrahowanie wiedzy z użyciem algorytmu AQ odbywa się w każdym kroku. W związku z tym czas działania algorytmu sumarycznie jest większy od czasu działania AQ-P1. Jednak z drugiej strony możliwość skrócenia okna przesuwnego w AQ-P2 zmniejsza czas indukowania reguł.

Rozdział 5

Ekstrakcja wiedzy ze strojonym modelem

Rozdział zawiera oryginalne rezultaty pracy, tzn. algorytm ekstrakcji wiedzy regułowej ze strojonym modelem przy użyciu struktury grafu do agregacji danych.

5.1 Wprowadzenie

Zaproponowane algorytmy w poprzednim rozdziale umożliwiają prospektywne podejmowanie decyzji oraz zmniejszają czas potrzebny do indukcji reguł. Tym niemniej, metody te zbytnio dopasowują się do danych (ang. *overfitting*). Stosowanie okna przesuwnego prowadzi do indukcji reguł dla niewielkiej przestrzeni wszystkich obserwacji, przez co wykorzystanie wiedzy w podejmowaniu decyzji może prowadzić do popełniania wielu błędów.

W niniejszym rozdziale proponuje się odmienne podejście do dotychczas zaprezentowanych metod indukcji reguł. Prezentowana metoda opiera się na idei algorytmów ze strojonym modelem, w których aktualizacja wiedzy odbywa się poprzez przetwarzanie pojedynczych obserwacji. Na proponowany algorytm składają się następujące kroki:

1. Obserwacje dla każdej klasy agregowane są za pomocą wag zmodyfikowanych grafów przepływów.

2. Mechanizm zapominania opiera się na zapominaniu wykładniczym na wagach grafów.

3. Przestrzeń przeszukiwań reguł jest zdefiniowana za pomocą grafu i ze względu na jego konstrukcję jest ona ograniczona.

4. Sposób wyznaczania przeszukiwań reguł uwzględnia *regularyzację*, która przeciwdziała zbytniemu dopasowaniu się do danych.

Uwaga. W kolejnych punktach algorytm ekstrakcji reguł z wykładniczym zapominaniem przedstawiony zostanie dla przypadku dwuklasowego, tzn. $y \in \{0, 1\}$. Następnie, w punkcie 5.2.4 zaprezentowano uogólnienie dla wielu klas.

5.2 Algorytm GRI

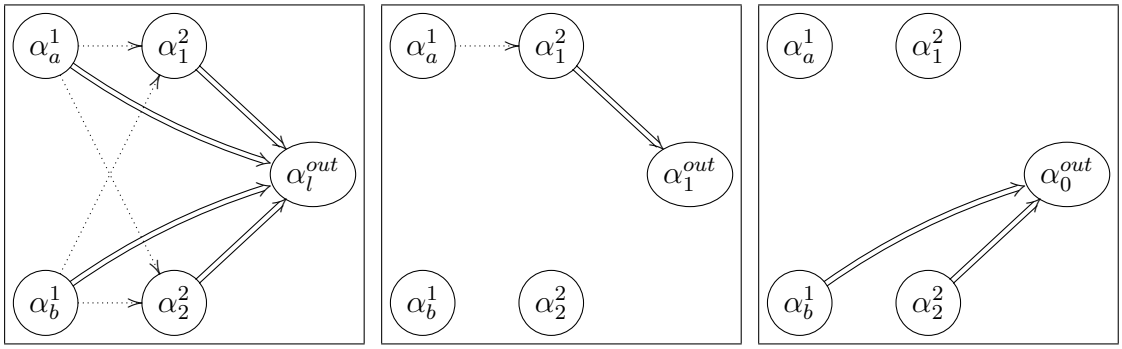
5.2.1 Reprezentacja reguł za pomocą grafu

Zanim przedstawimy szczegóły algorytmu, zastanówmy się najpierw nad konstrukcją reguły. Otóż pojedyncza reguła składa się z warunku, który jest w CNF, oraz decyzji. Zatem warunek jest koniunkcją wejściowych formuł elementarnych, natomiast decyzja jest pojedynczą wyjściową formułą elementarną. Ponadto, warto zaznaczyć, iż operator logiczny $\wedge$ jest łączny i przemienny, więc możemy przyjąć, że formuły wejściowe zawsze będziemy porządkowali. Wówczas pojedynczą regułę można przedstawić jako graf, w którym krawędzie łączące formuły wejściowe oznaczać będą operator $\wedge$, natomiast krawędź między formułą wejściową a wyjściową – $\Rightarrow$. Poza tym, ponieważ ustalimy kolejność formuł wejściowych, zamiast krawędzi mówić będziemy o łukach.

Przykład 5.2.1. Rozważmy obiekt o dwóch wejściach, $u^1 \in \{a, b\}$, $u^2 = \{1, 2\}$, oraz wyjściu, $y \in \{0, 1\}$. Zatem mamy sześć formuł elementarnych, α_a^1, α_b^1, α_1^2, α_2^2, α_0^{out}, α_1^{out}.
Dalej załóżmy, że wiedza o obiekcie wyrażona jest w następujący sposób:

$$\phi_1 = \left(\alpha_a^1 \wedge \alpha_1^2 \Rightarrow \alpha_1^{out}\right), \;\; \phi_2 = \left(\alpha_b^1 \Rightarrow \alpha_0^{out}\right), \;\; \phi_3 = \left(\alpha_2^2 \Rightarrow \alpha_0^{out}\right).$$

Wtedy wszystkie możliwe reguły reprezentowane przez graf zaznaczono na rysunku 5.1a, natomiast reguły dla klasy 1 pokazano na rysunku 5.1b, a dla klasy 0 – na rysunku 5.1c. Łuki odpowiadające operatorowi logicznemu koniunkcji, $\wedge$, zaznaczono linią kropkowaną, zaś między formułami wejściowymi a wyjściowymi – linią podwójną. Wówczas reguła dla klasy ϕ_1 pokazana jest na rysunku 5.1b. Koniunkcja formuł α_a^1 i α_1^2 jest wyrażona poprzez łuk z linią kropkowaną, zaś implikacja linią podwójną. Podobnie oznaczone są reguły ϕ_2 oraz ϕ_3.



(a) Wszystkie możliwe reguły. (b) Reguły dla klasy 1. (c) Reguły dla klasy 0.

Rysunek 5.1: Reguły reprezentowane przez graf dla przykładu 5.2.1.

Zatem każdy model regułowy może być wyrażony za pomocą grafu [40]

$$\mathcal{G} = (V, E), \tag{5.1}$$

gdzie V oznacza zbiór węzłów (wierzchołków), tj. formuł wejściowych i wyjściowych, zaś E oznacza zbiór łuków.

Warto zauważyć, że zbiór węzłów odpowiada zbiorowi wszystkich formuł elementarnych $\mathscr{A}$, natomiast zbiór wszystkich dopuszczalnych ścieżek w grafie dla każdej klasy odpowiada zbiorowi wszystkich reguł. Dalej wierzchołki będziemy oznaczać tak samo jak formuły elementarne.

Następnie określmy graf dla pozytywnych obserwacji, tj. dla $y = 1$, jako $\mathcal{G}_1$, natomiast dla negatywnych, tj. dla $y = 0$, jako $\mathcal{G}_0$. Zbiór wierzchołków jest taki sam dla obu przypadków. Ponadto przez $\mathbf{w}_1$ oznaczmy wagi związane z łukami w grafie pozytywnym, zaś

przez $\mathbf{w}_0$ – wagi w grafie negatywnym. Wagi określają liczbę wystąpień dwóch węzłów (formuł elementarnych) pośród obserwacji pozytywnych w grafie $\mathcal{G}_1$ oraz obserwacji negatywnych w $\mathcal{G}_0$. Dodatkowo, liczbę obserwacji negatywnych oraz pozytywnych oznaczamy odpowiednio przez w_0^{out} i w_1^{out}.

Jak już wspomniano, ze względu na łączność i przemienność operatora logicznego koniunkcji, $\wedge$, wystarczy przyjąć odpowiedni porządek wejść i używać łuków zamiast krawędzi, oraz zauważyć, że nie dopuszczamy możliwości łączenia wartości w ramach jednego wejścia. W związku z tym rozpatrujemy zorientowany, acykliczny, warstwowy graf. Pojedyncza warstwa grafu odpowiada jednemu wejściu, w której znajdują się formuły elementarne przyjmujące wszystkie wartości dla jednego wejścia. Na przykład na rysunku 5.1 warstwa pierwsza odpowiada wejściu $d = 1$, czyli węzły α_a^1, α_b^1, warstwa druga - wejściu $d = 2$ z węzłami α_1^2, α_2^2 oraz warstwa wyjściowa z α_l^{out}. Ponadto ustalmy porządek wejść wg liczności zbioru wartości od najmniejszej do największej, czyli otrzymujemy porządek $K_{(1)} \leqslant K_{(2)} \leqslant \ldots \leqslant K_{(D)}$, gdzie numer indeksu w nawiasie oznacza uporządkowane wejście. Ostatnia warstwa, tj. $(D+1)$-wsza, zawiera wyłącznie formułę wyjściową.

Łuk łączący dwa wierzchołki, tj. i-ty węzeł w s-tej warstwie i j-ty węzeł w t-tej warstwie jest oznaczony przez $e_{i,j}^{s\to t}$, np. łuk z α_b^1 do α_1^2 oznaczamy przez $e_{b,1}^{1\to 2}$, zaś łuk z α_b^1 do α_0^{out} – $e_{b,0}^{1\to out}$. Jak wspomniano wcześniej, z każdym łukiem wiążemy wagę, którą dla pozytywnego grafu opisujemy przez $w_{1,i,j}^{s\to out}$, a dla negatywnego – $w_{0,i,j}^{s\to out}$.

Zazwyczaj graf jest wyrażany za pomocą tzw. macierzy sąsiedztwa (ang. *adjacency matrix*) [40], która określa połączenia między węzłami, tzn. tworzona jest macierz $D \times D$, w której występowanie łuku między węzłami oznaczane jest 1, zaś brak – 0. Jednak korzystając z faktu, że graf reprezentujący model regułowy jest warstwowy, zorientowany oraz acykliczny, okazuje się, że wystarczy pamiętać mniej wag niż K^2. Następujący lemat określa liczbę łuków wystarczających do opisania grafu reprezentującego model regułowy.

Lemat 5.1. *W warstwowym, zorientowanym i acyklicznym grafie reprezentującym model regułowy istnieje* $\sum_{d=1}^{D-1} K_d \cdot \left(\sum_{\Delta=d+1}^{D} K_\Delta \right) + K$ *łuków.*

Dowód. Załóżmy, że d-ta warstwa zawiera K_d węzłów, $d = 1, 2, \ldots, D$. Wtedy dla każdego wierzchołka w pierwszej warstwie istnieją łuki do wszystkich pozostałych węzłów w warstwach $d > 1$ oraz łuk do węzła ostatniego (wyjściowego). Zatem dla pierwszej warstwy

mamy $K_1 \sum_{\Delta=2}^{D} K_\Delta + K_1$ łuków. Podobnie, dla drugiej warstwy istnieje $K_2 \sum_{\Delta=3}^{D} K_\Delta + K_2$ łuków, ponieważ niedopuszczalne są łuki do pierwszej warstwy. Analogicznie można podać liczbę łuków dla każdej warstwy $d < D$. Dla ostatniej warstwy istnieją tylko łuki do węzła wyjściowego, tzn., że istnieje K_D łuków.

Zatem ostatecznie otrzymujemy $\sum_{d=1}^{D-1} K_d \Big(\sum_{\Delta=d+1}^{D} K_\Delta + 1 \Big) + K_D$ łuków. Wyrażenie to można uprościć do następującej postaci

$$\sum_{d=1}^{D-1} K_d \Big(\sum_{\Delta=d+1}^{D} K_\Delta + 1 \Big) + K_D = \sum_{d=1}^{D-1} K_d \Big(\sum_{\Delta=d+1}^{D} K_\Delta \Big) + (K - K_D) + K_D =$$

$$= \sum_{d=1}^{D-1} K_d \Big(\sum_{\Delta=d+1}^{D} K_\Delta \Big) + K.$$

□

W związku z lematem 5.1 zamiast przetrzymywania K^2 parametrów w pamięci wystarczy zachowywać

$$\kappa = \sum_{d=1}^{D-1} K_d \cdot \Big(\sum_{\Delta=d+1}^{D} K_\Delta \Big) + K.$$

Dlatego też każdy graf reprezentujący model regułowy zawiera κ parametrów i może być zapisany w następującej postaci:

$$\mathrm{code}(\mathcal{G}) = \Big[\mathrm{code}(warstwa_1) \ \ \mathrm{code}(warstwa_2) \ \ \ldots \ \ \mathrm{code}(warstwa_D) \Big], \tag{5.2}$$

gdzie $\mathrm{code}(warstwa_d)$ jest ciągiem 0-1 (kodem) dla d-tej warstwy, $d = 1, 2, \ldots, D$,

$$\mathrm{code}(warstwa_d) =$$

$$\Big[e_{1,1}^{d\to d+1} \ e_{1,2}^{d\to d+1} \ldots e_{1,k_{d+1}}^{d\to d+1} \ e_{1,1}^{d\to d+2} \ldots e_{1,k_{d+2}}^{d\to d+2} \ldots e_{1,1}^{d\to D} \ldots e_{1,k_D}^{d\to D} \ e_{1,l}^{d\to out}$$

$$e_{2,1}^{d\to d+1} \ e_{2,2}^{d\to d+1} \ldots e_{2,k_{d+1}}^{d\to d+1} \ e_{2,1}^{d\to d+2} \ldots e_{2,k_{d+2}}^{d\to d+2} \ldots e_{2,1}^{d\to D} \ldots e_{2,k_D}^{d\to D} \ e_{2,l}^{d\to out}$$

$$\ldots$$

$$e_{k_d,1}^{d\to d+1} \ e_{k_d,2}^{d\to d+1} \ldots e_{k_d,k_{d+1}}^{d\to d+1} \ e_{k_d,1}^{d\to d+2} \ldots e_{k_d,k_{d+2}}^{d\to d+2} \ldots e_{k_d,1}^{d\to D} \ldots e_{k_d,k_D}^{d\to D} \ e_{k_d,l}^{d\to out} \Big]$$

Na przykład dla grafu podanego na rysunkach 5.1b i 5.1c kody są następujące:

$$\mathrm{code}(\mathcal{G}_1) = [100\ 000\ 1\ 0],$$

$$\mathrm{code}(\mathcal{G}_0) = [000\ 001\ 0\ 1].$$

Przykładowo pierwszy kod mówi, że formuła atomowa α_a^1 jest połączona z α_1^2 i nie jest połączona z α_2^2 oraz α_1^{out}. Ponadto, dla formuły α_b^1 nie ma łuków do innych formuł, natomiast z warstwy drugiej istnieje jedno połączenie między α_1^2 i α_1^{out}.

Dalej zakładać będziemy, że zarówno grafy $\mathcal{G}_1, \mathcal{G}_0$, jak również ich wagi $\mathbf{w}_1, \mathbf{w}_0$, są zakodowane przy pomocy (5.2).

5.2.2 Uczenie i ekstrakcja reguł

Jak wskazano wcześniej, każda obserwacja jest najbardziej uszczegółowioną regułą, tzn. zawiera po jednej formule atomowej dla każdego wejścia. Zatem każda obserwacja może być zakodowana przy użyciu (5.2). Na przykład, jak dla obiektu przedstawionego w 5.2.1, obserwacja postaci $\mathbf{u} = (\alpha_a^1 \wedge \alpha_1^2)$ jest przedstawiona za pomocą kodu jako $code(\mathbf{u}) = [10000010]$. Co więcej, wagi w grafach odpowiadają liczbie wystąpień każdej pary formuł wejściowych oraz wyjściowej w obserwacji. Dlatego, posiadając ciąg uczący, można uaktualniać wagi grafów niezależnie od kolejności obserwacji oraz w sposób przyrostowy. Jeśli n-ta obserwacja jest postaci $(\mathbf{u}_n, y_n)$, to uaktualnianie wag odbywa się wg następujących wyrażeń:

$$\mathbf{w}_{y_n} := \mathbf{w}_{y_n} + \mathrm{code}(\mathbf{u}_n) \tag{5.3}$$

$$w_{y_n}^{out} := w_{y_n}^{out} + 1, \tag{5.4}$$

Procedura uaktualniania wag może być wyrażona w podany sposób, ponieważ długości kodów wag oraz obserwacji są identyczne oraz odpowiadają tym samych formułom wejściowym.

Dodatkowo warto zauważyć, że przy takim postępowaniu uaktualniania wag można odtworzyć ciąg uczący, ale bez zachowania kolejności pojawienia się obserwacji. Procedura postępowania jest następująca. Zaczynając od pierwszego węzła w pierwszej warstwie należy przejść przez wszystkie pozostałe warstwy, odwiedzając tylko jeden węzeł w warstwie, aż do węzła końcowego. Przejście między węzłami jest możliwe wtedy, gdy waga jest dodatnia. Natomiast po przejściu wagę na łuku należy pomniejszyć o 1. Postępując w ten

sposób dla wszystkich węzłów w pierwszej warstwie zostanie odtworzony pierwotny ciąg uczący.

Przykład 5.2.2. Dla obiektu z przykładu 5.2.1 rozpatrzmy następujące dane:

$$\left\{(\begin{bmatrix} a \\ 1 \end{bmatrix}, 1), (\begin{bmatrix} a \\ 2 \end{bmatrix}, 0), (\begin{bmatrix} b \\ 1 \end{bmatrix}, 0), (\begin{bmatrix} b \\ 2 \end{bmatrix}, 0)\right\}$$

Wówczas wagi są następujące

$$\mathbf{w}_1 = [101\ 000\ 1\ 0],$$
$$\mathbf{w}_0 = [011\ 112\ 1\ 2],$$
$$w_1^{out} = 1,$$
$$w_0^{out} = 3.$$

Następnie, aby ocenić pary węzłów i ostatecznie – ścieżek z końcem w wierzchołku końcowym, czyli reguł, należy określić odpowiednie kryterium. W uczeniu maszynowym, jak również w zadaniu ekstrakcji reguł, zazwyczaj korzysta się z miar *pokrycia* (ang. *coverage*) oraz *dokładności*[1](ang. *accuracy*) [63, 117]. *Pokrycie* jest miarą mówiącą o stopniu ogólności reguły, natomiast *dokładności* jest miarą wyrażającą stopień specjalizacji reguły. Na przykład, reguła $\alpha_a^1 \Rightarrow \alpha_1^{out}$ jest bardziej ogólna od $\alpha_a^1 \wedge \alpha_1^2 \Rightarrow \alpha_1^{out}$.

Zanim wprowadzimy odpowiednie definicje, określmy zbiór obserwacji ze zbioru uczącego, które są pokryte przez warunek reguły ϕ jako $\mathcal{E}_{\phi_{in}}$, natomiast zbiór obserwacji ze zbioru uczącego w klasie $y = l$ jako $\mathcal{E}_l$. Wówczas miara pokrycia przyjmuje postać

$$\mu_c(\phi_{in}, l) = \frac{\text{card}\{\mathcal{E}_{\phi_{in}} \cap \mathcal{E}_l\}}{\text{card}\{\mathcal{E}_l\}}, \tag{5.5}$$

zaś miara dokładności wyraża się za pomocą następującej równości

$$\mu_a(\phi_{in}, l) = \frac{\text{card}\{\mathcal{E}_{\phi_{in}} \cap \mathcal{E}_l\}}{\text{card}\{\mathcal{E}_{\phi_{in}}\}}. \tag{5.6}$$

W przypadku, gdy powyższe wyrażenia są nieoznaczone, to przyjmujemy, że są równe zero.

Następnie warto zauważyć, że *dokładność* może być wyrażona poprzez *pokrycie* [117]

$$\mu_a(\phi_{in}, l) = \frac{\mu_c(\phi_{in}, l)\ \text{card}\{\mathcal{E}_l\}}{\mu_c(\phi_{in}, 0)\ \text{card}\{\mathcal{E}_0\} + \mu_c(\phi_{in}, 1)\ \text{card}\{\mathcal{E}_1\}}. \tag{5.7}$$

[1]*Dokładność* czasem zwana jest *zaufaniem* (ang. *confidence*) [63] lub *pewnością* (ang. *certainty*) [117].

Co więcej, warto zaznaczyć, że dla dowolnej pary formuł wejściowych A i B, μ_c jest antymonotoniczna, tj.

$$\mu_c(A, y) \geqslant \mu_c(A \wedge B, y), \tag{5.8}$$

zaś μ_a jest monotoniczna, tzn.

$$\mu_a(A, y) \leqslant \mu_a(A \wedge B, y). \tag{5.9}$$

Obie własności wynikają bezpośrednio z definicji. Zbiór pokrytych obserwacji przez pojedynczą formułę A może być taki sam, albo większy niż zbiór dla $A \wedge B$, formalnie $\mathrm{card}\{\mathcal{E}_A\} \geqslant \mathrm{card}\{\mathcal{E}_{A\wedge B}\}$. Zatem, przy dodaniu formuły licznik w wyrażeniu na *pokrycie* może zmaleć, zaś mianownik jest stały, a więc wartość *pokrycia* może zmaleć lub pozostać bez zmian. Jednak w przypadku *dokładności* zarówno licznik jak i mianownik mogą zmaleć, ale mianownik maleje szybciej niż licznik, ponieważ $\mathrm{card}\{\mathcal{E}_{A\wedge B} \cap \mathcal{E}_y\} \leqslant \mathrm{card}\{\mathcal{E}_{A\wedge B}\}$. Dlatego wartość *dokładności* może wzrosnąć lub pozostać bez zmian.

Miary *pokrycia* i *dokładności* są odpowiednie do mierzenia, osobno, stopnia generalizacji i specjalizacji reguły. Natomiast w problemie ekstrakcji reguł należy znaleźć taką wiedzę, która jest dokładną generalizacją obserwacji [106]. W związku z tym należy zaproponować miarę, która uwzględni obie wielkości. W tym celu proponowane jest następujące kryterium syntetyczne, które jest kombinacją wypukłą *pokrycia* i *dokładności*,

$$q(\phi_{in}, l) = \beta\, \mu_c(\phi_{in}, l) + (1 - \beta)\, \mu_a(\phi_{in}, l), \tag{5.10}$$

gdzie $\beta \in [0, 1]$ i określa gdzie znajduje się środek ciężkości pomiędzy oceną generalizacji, wyrażoną przez μ_c, i specjalizacji, wyrażoną przez μ_a.

Zatem, posiadając ciąg uczący, a co za tym idzie – wagi grafów $\mathcal{G}_0$ i $\mathcal{G}_1$, można policzyć *pokrycie* i *dokładność* dla każdej krawędzi $e_{i,j}^{s\to t}$ w l-tej klasie. Oznaczając przez $w_{l,i,j}^{s\to t}$ wagę łuku $e_{i,j}^{s\to t}$ w klasie $y = l$, oraz przez w_l^{out} liczbę obserwacji z klasy $y = l$, *pokrycie* wyznaczamy w następujący sposób.

$$\mu_c(e_{i,j}^{s\to t}, l) = \frac{w_{l,i,j}^{s\to t}}{w_l^{out}}, \tag{5.11}$$

W celu policzenia *dokładności* stosujemy (5.7), czyli otrzymujemy

$$\mu_a(e_{i,j}^{s\to t}, l) = \frac{\mu_c(e_{i,j}^{s\to t}, l)\; w_l^{out}}{\mu_c(e_{i,j}^{s\to t}, 0)\; w_0^{out} + \mu_c(e_{i,j}^{s\to t}, 1)\; w_1^{out}}. \tag{5.12}$$

Każda ścieżka kończąca się w węźle wyjściowym jest równoważna regule, dlatego istotne jest określenie jakości ścieżki. Przez ścieżkę π o długości mniejszej lub równej D rozumiemy ciąg niepowtarzających się węzłów o początku w $\alpha_{k_d}^d$ i końcu w α_l^{out} taki, że pomiędzy dowolnymi dwoma sąsiadującymi węzłami istnieje łuk je łączący. Każdy węzeł w ścieżce należy do osobnej warstwy. Zatem, aby policzyć wartość kryterium dla ścieżki skorzystamy z antymonotoniczności miary μ_c, wówczas

$$\mu_c(\pi, l) = \min_{e \in \pi} \{\mu_c(e, l)\}. \tag{5.13}$$

Następnie, korzystając z (5.7) *dokładność* może być wyznaczona w podany sposób:

$$\mu_a(\pi, l) = \frac{\mu_c(\pi, l)\ w_l^{out}}{\mu_c(\pi, 0)\ w_0^{out} + \mu_c(\pi, 1)\ w_1^{out}}. \tag{5.14}$$

Posiadając wagi dla grafu pozytywnego i negatywnego oraz określoną wartość parametru β można policzyć wartość kryterium (5.10) dla dowolnej ścieżki w grafie reprezentującym model regułowy. W związku z tym można wygenerować wszystkie możliwe ścieżki w grafie pozytywnym i negatywnym o $q > 0$ dla każdej klasy. Wówczas otrzymalibyśmy 2^K reguł. Jednak taka wiedza jest nieakceptowalna, ponieważ reguły odzwierciedlałyby pojedyncze obserwacje oraz część reguł wzajemnie pokrywałaby się. Poza tym, przestrzeń przeszukiwań jest zbyt duża i przez to takie podejście jest zbyt kosztowne obliczeniowo. Dlatego też potrzebna jest procedura, która pozwoli ograniczyć przestrzeń dopuszczalnych rozwiązań.

Pomysł ograniczenia przestrzeni dopuszczalnych rozwiązań jest oparty na zdefiniowaniu nowego grafu, w którym każda krawędź jest albo pozytywna, albo negatywna. Wówczas ścieżka jest dopuszczalna, jeśli wszystkie krawędzie do niej należące są jednego znaku. Najpierw wyliczane są wagi grafów negatywnego i pozytywnego, tj. $\mathbf{w}_0, \mathbf{w}_1$, oraz w_0^{out}, w_1^{out}. Następnie dla każdego łuku w obu grafach wyliczana jest wartość kryterium (5.10). Skutkuje to w otrzymaniu nowych grafów negatywnego i pozytywnego o wagach odpowiednio $\mathbf{q}_0, \mathbf{q}_1$. W ostatnim kroku liczona jest różnica między $\mathbf{q}_1$ i $\mathbf{q}_0$, którą oznaczamy $\mathbf{q}$. Tak otrzymany graf, w którym krawędzie są dodatnie lub ujemne, określa przestrzeń dopuszczalnych rozwiązań. Procedura wyznaczania takiego grafu przedstawiona jest w algorytmie 5.2.1.

Algorytm 5.2.1. Ograniczenie przestrzeni dopuszczalnych rozwiązań.

Wejście: (i) ciąg uczący $\mathcal{D}$, (ii) β, (iii) kryterium jakości (5.10), (iv) *pokrycie* (5.11), (v) *dokładność* (5.12).

Wyjście: Graf determinujący przestrzeń dopuszczalnych rozwiązań.

Krok 1: Korzystając z ciągu uczącego $\mathcal{D}$ policz $\mathbf{w}_1, \mathbf{w}_0, w_0^{out}, w_1^{out}$.

Krok 2: Dla zadanej wartości β, dla każdego łuku w grafie $\mathcal{G}_1$ oraz $\mathcal{G}_0$ policz wartość kryterium jakości (5.10) korzystając z (5.11) i (5.12). Wartości te oznacz przez $\mathbf{q}_1$ i $\mathbf{q}_0$ dla $\mathcal{G}_1$ and $\mathcal{G}_0$, odpowiednio. (Oba wektory są rozmiarów κ).

Krok 3: Policz różnicę między $\mathbf{q}_1$ i $\mathbf{q}_0$, tj.

$$\mathbf{q} = \mathbf{q}_1 - \mathbf{q}_0. \tag{5.15}$$

Wektor $\mathbf{q}$ reprezentuje kod grafu (przestrzeni dopuszczalnych rozwiązań) z wagami na łukach, które są pozytywne lub negatywne. Innymi słowy, graf zakodowany za pomocą $\mathbf{q}$ zawiera krawędzie, które są zaklasyfikowane albo do klasy 1 albo 0.

Zatem, posiadając zdefiniowaną przestrzeń przeszukiwań za pomocą $\mathbf{q}$ nie wszystkie ścieżki są osiągalne. Po pierwsze ścieżki, które zawierają zarówno krawędzie ze znakiem dodatnim jak i ujemnym są uznawane za niedopuszczalne. Po drugie wyłącznie te ścieżki mogą być rozpatrywane, dla których wartość kryterium q jest większa od zadanego progu $\epsilon \in [0, 1)$. Po określeniu przestrzeni przeszukiwań można zaproponować algorytm indukcji reguł, zwany dalej ***G**raph-based **R**ules **I**nducer* (GRI). Wartość kryterium (5.10) dla krawędzi $e_{i,j}^{s \to t}$ oznaczać będziemy $q_{i,j}^{s \to t}$.

Algorytm 5.2.2. Graph-based Rules Inducer.

Wejście: (i) ciąg uczący $\mathcal{D}$, (ii) β, (iii) wartość kryterium (5.10), (iv) *pokrycie* (5.11), (v) *dokładność* (5.12), (vi) ϵ.

Wyjście: Model regułowy Φ.

Krok 1: Posiadając obserwacje $\mathcal{D}$ i β zastosuj algorytm 5.2.1, żeby otrzymać $\mathbf{q}$. Ustaw $\mathcal{P}_+ = \emptyset$, $\mathcal{P}_- = \emptyset$, oraz $d := D$.

Krok 2: Dla każdego $i = 1, \ldots, K_d$ rozważ ścieżkę $\pi = (e_{i,y}^{d \to out})$. Jeśli $q_{i,y}^{d \to out} > 0$, to $\mathcal{P}_+ := \mathcal{P}_+ \cup \{\pi\}$. W przeciwnym razie, jeśli $q_{i,y}^{d \to out} < 0$, to $\mathcal{P}_- := \mathcal{P}_- \cup \{\pi\}$.

Krok 3: Dla każdej $\pi \in \mathcal{P}_+$ takiej, że $\pi = (e_{j,\cdot}^{b \to \cdot} \ldots)$ i dla wszystkich $i = 1, \ldots, K_d$ rozważ ścieżkę $\overline{\pi} = (e_{i,j}^{d \to b}\, e_{j,\cdot}^{b \to \cdot} \ldots)$. Jeśli $q_{i,j}^{d \to b} > 0$, to $\mathcal{P}_+ := \mathcal{P}_+ \cup \{\overline{\pi}\}$. Dla każdej $\pi \in \mathcal{P}_-$ takiej, że $\pi = (e_{j,\cdot}^{b \to \cdot} \ldots)$ i dla każdego $i = 1, \ldots, K_d$ rozważ ścieżkę $\overline{\pi} = (e_{i,j}^{d \to b}\, e_{j,\cdot}^{b \to \cdot} \ldots)$. Jeśli $q_{i,j}^{d \to b} < 0$, to $\mathcal{P}_- := \mathcal{P}_- \cup \{\overline{\pi}\}$.

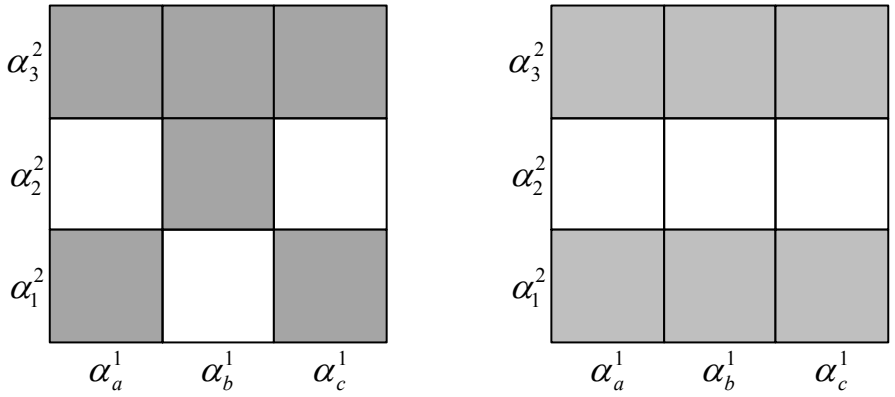
Krok 4: Jeśli $d > 1$, to $d := d - 1$ i idź do kroku 2.

Krok 5: Dla każdej $\pi \in \mathcal{P}_- \cup \mathcal{P}_+$. Jeśli $q(\pi, 1) > \epsilon$ lub $q(\pi, -1) > \epsilon$, to formułuj regułę ϕ związaną ze znakiem ścieżki P i uaktualnij model $\Phi := \Phi \vee \phi$.

W algorytmie 5.2.2 najpierw wyznaczany jest graf określający przestrzeń dopuszczalnych rozwiązań. $\mathcal{P}_+$ i $\mathcal{P}_-$ oznaczają zbiory wszystkich możliwych ścieżek w $\mathbf{q}$, które są, odpowiednio, o znaku dodatnim lub ujemnym. W celu wyznaczenia znalezienia wyłącznie tych ścieżek, które mogą być regułami, tj. ścieżka ma jeden znak na wszystkich krawędziach, algorytm działa od tyłu, tzn. od ostatniej warstwy do pierwszej. Pomimo że liczność obu zbiorów może rosnąć wykładniczo ze względu na liczbę wejść, w wielu praktycznych zastosowaniach, tj. gdy liczba wejść nie jest zbyt duża (np. $D < 10$), liczności te wydają się być akceptowalne. Natomiast w przypadku, gdy liczba wejść jest zbyt duża, wówczas należy zaproponować heurystykę pozwalającą na wyeliminowanie krawędzi, które wydają się być nieprzydatne w formułowaniu reguł. Jakkolwiek, w sytuacjach, w których dokładność wiedzy jest niezwykle istotna, np. w zastosowaniach medycznych, gdy zdrowie człowieka jest przedmiotem ekstrakcji wiedzy, należy stosować z rozwiązania dokładnego, nawet kosztem dużej złożoności obliczeniowej.

Uwagi:

1. Grafy $\mathcal{G}_0$ oraz $\mathcal{G}_1$ mogą być postrzegane jako grafy przepływów Pawlaka [117, 118], ale o nieco odmiennej strukturze. W prezentowanym podejściu każdy węzeł dotyczy pojedynczej formuły atomowej, a nie koniunkcji formuł jak w oryginalnym podejściu Pawlaka. Ponadto, dopuszczamy tutaj krawędzie między każdą formułą wejściową a wyjściową, czego nie ma w założeniach grafów przepływów. Tym niemniej, wagi związane z łukami można postrzegać jako przepływ informacji między warstwami.



(a) Oryginalny opis. (b) Opis po zastosowaniu algorytmu 5.2.1.

Rysunek 5.2: Przykład zastosowania algorytmu 5.2.1.

Poza tym, ilość informacji wpływającej do d-tej warstwy w klasie $y = l$, $f_{in}(d, l)$ równa się wartości wypływającej, $f_{out}(d, l)$, tj.

$$f_{in}(d,l) = \frac{1}{d} \sum_{s=1}^{d-1} \sum_{i=1}^{K_s} \sum_{j=1}^{K_d} w_{l,i,j}^{s \to d}, \tag{5.16}$$

$$f_{out}(d,l) = \frac{1}{D+1-d} \sum_{t=d+1}^{D+1} \sum_{i=1}^{K_d} \sum_{j=1}^{K_t} w_{l,i,j}^{d \to t}, \tag{5.17}$$

Ze względu na sposób uaktualniania wag (5.3) wynika, że $f_{in}(d, l) = f_{out}(d, l)$. Równania (5.16) i (5.17) mogą być postrzegane jako równania zachowania przepływu [118].

2. Ograniczenie przestrzeni dopuszczalnych rozwiązań zastosowany w algorytmie GRI ma znaczenie na dwójnasób. Po pierwsze, tylko ścieżki jednoznakowe są dopuszczalne. Po drugie, nie wszystkie modele są osiągalne przy takiej reprezentacji. Aby to zobrazować rozpatrzmy przykład jak na rysunku 5.2. Obiekt opisany jest dwoma wejściami i każde wejście posiada trzy wartości. Klasa 1 jest oznaczona kolorem szarym, natomiast klasa 0 – białym (szare i białe prostokąty na rysunku 5.2a). Po zastosowaniu algorytmu 5.2.1 niemożliwe jest, aby osiągnąć reguły $\alpha_b^1 \wedge \alpha_2^2 \Rightarrow \alpha_1^{out}$

i $\alpha_a^1 \wedge \alpha_2^2 \Rightarrow \alpha_0^{out}$, ponieważ oznaczałoby to, że łuk w grafie $\mathbf{q}$ pomiędzy α_2^2 i węzłem wyjściowym musiałby być jednocześnie pozytywny i negatywny. Dlatego, w zależności od ciągu uczącego, wiedza może przyjąć postać taką, jak na rysunku 5.2b.

Zatem ograniczenie przestrzeni dopuszczalnych rozwiązań jest rodzajem regularyzacji, dzięki czemu model staje się odporny na szum oraz zbytnie dopasowanie do danych (*overfitting*). Oczywiście w przypadku, gdy rzeczywisty obiekt jest w postaci jak na rysunku 5.2a, to wiedza będzie mniej dokładna. Jednakże, w przypadku losowym i zmiennego kontekstu lepiej, aby wiedza była odporna na zbytnie dopasowanie do danych, nawet kosztem dokładności.

3. Graf $\mathbf{q}$ może być użyty nie tylko do ograniczenia przestrzeni dopuszczalnych rozwiązań, ale również jako klasyfikator. Wówczas procedura klasyfikacji nowo pojawiającej się obserwacji jest bardzo prosta. Najpierw należy znaleźć wszystkie jednoznakowe podścieżki obserwacji[2] (obserwacja traktowana jest jako ścieżka). Następnie wybierana jest ścieżka o największej wartości kryterium q. Znak dla tak wybranej podścieżki zwracany jest jako klasa.

   W przypadku, gdy graf $\mathbf{q}$ wykorzystany zostanie jako klasyfikator, to będziemy mówili o klasyfikatorze GRI. Warto zaznaczyć, że klasyfikator taki uczy się w czasie wielomianowym, natomiast proces klasyfikacji, dla dużego D, jest etapem kosztownym (wykładniczym). W praktyce, podczas sprawdzania podścieżek, gdy dwa łuki mają inny znak, podścieżka może zostać pominięta. Przyspieszy to łączny czas testowania, jednak problem nadal pozostaje wykładniczy.

4. Podane algorytmy ograniczania przestrzeni dopuszczalnych rozwiązań oraz GRI są algorytmami z uczeniem wsadowym. W kolejnym punkcie posłużą one jako podstawa do podejścia z mechanizmem zapominania i uaktualniania przyrostowego.

5.2.3 Algorytm GRI z mechanizmem zapominania

Przedstawiony algorytm GRI jest algorytmem ekstrakcji wiedzy z uczeniem przyrostowym, ponieważ wagi grafów są uaktualnianie w każdym momencie. Jednak w przypadku

[2]Dla obserwacji jako ścieżka o długości D istnieje $2^D - 1$ wszystkich możliwych podścieżek.

ze zmieniającym się kontekstem należy dodatkowo zaproponować mechanizm zapominania.

Stosowanie okna przesuwnego do uaktualniania wag grafów jest niepraktyczne, ponieważ po pierwsze należy przechowywać wszystkie obserwacje zawarte w oknie przesuwnym, a po drugie – wagi grafów muszą być uaktualnianie za każdym razem na nowo. Jednak posiadając wagi grafów można dokonać zapominania bezpośrednio na nich za pomocą zapominania wykładniczego (zapominania ze współczynnikiem zapominania). Wyrażenia na uaktualnianie wag (5.3) i (5.4) mogą być zmodyfikowane w następujący sposób, jeśli pojawiła się nowa obserwacja $(\mathbf{u}_{N+1}, y_{N+1})$ i $y_{N+1} = 1$, to

$$\mathbf{w}_1 := \gamma \mathbf{w}_1 + \mathrm{code}(\mathbf{u}_{N+1}), \tag{5.18}$$

$$\mathbf{w}_0 := \gamma \mathbf{w}_0, \tag{5.19}$$

$$w_1^{out} := \gamma w_1^{out} + 1, \tag{5.20}$$

$$w_0^{out} := \gamma w_0^{out} \tag{5.21}$$

lub jeśli $y_{N+1} = 0$, to

$$\mathbf{w}_1 := \gamma \mathbf{w}_1, \tag{5.22}$$

$$\mathbf{w}_0 := \gamma \mathbf{w}_0 + \mathrm{code}(\mathbf{u}_{N+1}), \tag{5.23}$$

$$w_1^{out} := \gamma w_1^{out}, \tag{5.24}$$

$$w_0^{out} := \gamma w_0^{out} + 1 \tag{5.25}$$

gdzie $\gamma \in [0, 1]$ oznacza współczynnik zapominania.

Tak zdefiniowany mechanizm zapominania w literaturze określa się jako zapominanie wykładnicze [21, 113]. Współczynnik zapominania γ jest zazwyczaj bliski 1 i okazuje się, że podejście takie może być postrzegane jako ważone okno przesuwne o rozmiarze równym $3/(1-\gamma)$. Wszystkie obserwacje, które są starsze niż rozmiar okna, tj. $3/(1-\gamma)$, wpływają na model z wagą mniejszą niż 0.05 [113].

Aby umożliwić uaktualniania wiedzy dla zmieniającego się kontekstu, należy zmodyfikować algorytm 5.2.1 oraz algorytm 5.2.2 poprzez dodanie zapominania wykładniczego.

Algorytm 5.2.3. Ograniczenie przestrzeni dopuszczalnych rozwiązań z zapominaniem.

Wejście: (i) ciąg uczący $\mathcal{D}$, (ii) β, (iii) kryterium jakości (5.10), (iv) *pokrycie* (5.11), (v) *dokładność* (5.12), (vi) γ, (vii) $\mathbf{w}_1, \mathbf{w}_0, w_1^{out}, w_0^{out}$ – wektory zerowe, (viii) $N := 0$.

Wyjście: Graf determinujący przestrzeń dopuszczalnych rozwiązań.

Krok 1: Ustaw $N := N+1$, pobierz obserwację z ciągu uczącego, $(\mathbf{u}_N, y_N)$. Jeśli obserwacja jest błędnie zaklasyfikowana i $y_N = 1$, to uaktualnij wagi grafów korzystając z (5.18) i (5.19). Jeśli obserwacja jest błędnie zaklasyfikowana oraz $y_N = 0$, to uaktualnij wagi grafów korzystając z (5.22) i (5.23). W przeciwnym razie uaktualnij wagi grafów za pomocą (5.3) i (5.4).

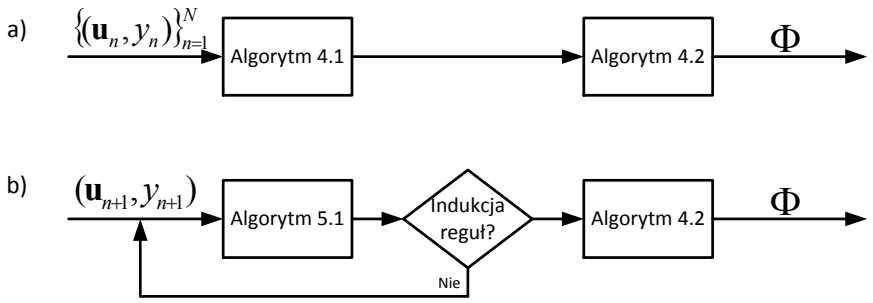
Krok 2 Dla zadanego β, dla każdego łuku w grafach $\mathcal{G}_1$ oraz $\mathcal{G}_0$ policz wartość kryterium jakości (5.10) korzystając z (5.11) i (5.12). Oznacz te wagi przez $\mathbf{q}_1$ oraz $\mathbf{q}_0$ odpowiednio dla $\mathcal{G}_1$ i $\mathcal{G}_0$.

Krok 3: Policz różnicę między $\mathbf{q}_1$ i $\mathbf{q}_0$, tj. (5.15).

Powyższa procedura jest modyfikacją algorytmu 5.2.1, w którym uaktualnianie wag grafów jest wykonywane przyrostowo i dodatkowo, w przypadku błędnie zaklasyfikowanej obserwacji[3], dokonywane jest zapominanie wykładnicze.

Różnicę między algorytmem GRI bez zapominania oraz z zapominaniem zaznaczono na rysunku 5.3. Na rysunku 5.3.a zaznaczono metodę bez zapominania, w której uaktualnianie wag grafów odbywa się przyrostowo, ale z wykorzystaniem całego ciągu uczącego. Natomiast na rysunku 5.3.b zaprezentowano algorytm z zastosowaniem mechanizmu zapominania wykładniczego, w którym obserwacje przetwarzane są pojedynczo i po pojawieniu się nowej wagi grafów są zapominanie ze współczynnikiem. Dodatkowo, w przypadku z zapominaniem, decyzja odnośnie liczby kroków po ilu należy uruchomić algorytm indukcji reguł (bloczek "Indukcja reguł" na rysunku 5.3.b) jest podejmowana przez użytkownika.

[3]Zapominanie można też stosować niezależnie od popełnionego błędu, tzn. w każdym momencie.



Rysunek 5.3: Schematy algorytmu GRI z ograniczeniem przestrzeni rozwiązań: a) bez zapominania, b) z zapominaniem wykładniczym.

5.2.4 Przypadek wieloklasowy

Na wstępie niniejszego rozdziału zaznaczyliśmy, że po prezentacji algorytmu dla przypadku o dwóch wartościach wyjść zostanie on uogólniony dla przypadku o liczbie klas $Y > 2$. W przypadku wieloklasowym zazwyczaj stosuje podejście *jeden przeciw wszystkim* (ang. *one-versus-the-rest*) [13]. Polega ono na tym, że dla każdej wartości wyjścia rozwiązuje się problem dwuklasowy.

Dla ekstrakcji wiedzy z użyciem reprezentacji grafowej dla każdej wartości wyjścia należy budować model, w którym graf pozytywny zawiera informacje o zadanej klasie, natomiast graf negatywny – o wszystkich pozostałych klasach. Na przykład dla $Y = 3$ otrzymujemy (indeks góry oznacza numer klasy, znak $+$ określa graf pozytywny, znak $-$ określa graf negatywny): $\mathcal{G}_+^0$, $\mathcal{G}_+^1$, $\mathcal{G}_+^2$ oraz $\mathcal{G}_-^0$, $\mathcal{G}_-^1$, $\mathcal{G}_-^2$, gdzie wagi dla $\mathcal{G}_-^0$ są sumą wag $\mathcal{G}_+^1$ i $\mathcal{G}_+^2$, wagi dla $\mathcal{G}_-^1$ są sumą wag $\mathcal{G}_+^0$ i $\mathcal{G}_+^2$, wagi dla $\mathcal{G}_-^2$ są sumą wag $\mathcal{G}_+^0$ i $\mathcal{G}_+^1$. Zatem wystarczy przechowywać wagi wyłącznie dla grafów pozytywnych, ponieważ grafy negatywne definiowane są przez grafy pozytywne z pozostałych klas.

Indukcja reguł w przypadku wieloklasowym odbywa się w ten sam sposób, jak przedstawiono w algorytmie GRI, tylko jest on powtarzany dla Y klas i bez wyznaczania reguł dla grafów negatywnych. Stosowanie grafu $\mathbf{q}$ dla każdej klasy do klasyfikiacji powoduje, że nowa obserwacja jest zawsze klasyfikowana do klasy o najwyżej wartości kryterium.

Rozdział 6

Badania empiryczne

6.1 Plan i zakres badań

W ramach niniejszego rozdziału zaprezentowano wyniki badań empirycznych z użyciem opracowanych i przedstawionych algorytmów wykrywania zmian oraz ekstrakcji wiedzy regułowej. Oprócz zilustrowania poprawności działania zaproponowanych algorytmów, eksperymenty miały na celu:

- sprawdzenie, czy proponowane algorytmy wykrywania zmian kontekstu są w stanie wykryć rzeczywiste momenty zmian, oraz porównanie wyników z algorytmami wykrywania zmian znanymi w literaturze przedmiotu;
- zbadanie wpływu długości okna oraz parametru wrażliwości na działanie algorytmów wykrywania zmian kontekstu;
- porównanie, ze względu na podane wcześniej kryteria (w przypadku deterministyczym – 1.7, w przypadku losowym – predykcyjny błąd sekwencyjny (1.14), prezentowanych w niniejszej pracy algorytmów ekstrakcji wiedzy z algorytmami z uczeniem przyrostowym znanymi w literaturze przedmiotu;
- zbadanie wpływu współczynnika zapominania, γ, oraz ważenia między generalizacją a specjalizacją, β, na wartość błędu w metodzie GRI.

Badania przeprowadzono dla:

- zadania wykrywania zmian kontekstu z wykorzystaniem rzeczywistego zbioru danych *Coal-mining distaster data* [69];
- zadania wykrywania zmian kontekstu w ocenie jakości działania systemu zorientowanego na usługi;
- zadania ekstrakcji wiedzy w przypadku deterministycznych z wykorzystaniem benchmarkowego zbioru danych *STAGGER* [134].
- zadania ekstrakcji wiedzy w przypadku losowym z wykorzystaniem rzeczywistego zbioru danych *Electricity* [59];
- zadania ekstrakcji wiedzy w systemie wspomagania przeprowadzenia wywiadu lekarskiego w terapii cukrzycy [19, 144, 148, 153] z wykorzystaniem rzeczywistego zbioru danych [153].

W kolejnych punktach przedstawiono szczegółowy opis badań wraz z krokami przeprowadzenia eksperymentu oraz omówieniem otrzymanych wyników.

6.2 Zadanie wykrywania zmian kontekstu – *Coal-mining distaster data*

Opis zbioru danych

Zbiór *Coal-mining distaster data*, który został pierwotnie przedstawiony w [69], jest standardowym zbiorem danych wykorzystywanym do testowania metod wykrywania zmian kontekstu. Zawiera on dane dotyczące 191 wypadków w kopalniach w Wielkiej Brytania od 1851 do 1962, w których zginęło dziesięciu lub więcej górników.

Obiektem w tym przypadku jest system kopalń w Wielkiej Brytanii, w którym obserwowanym **wyjściem** jest liczba wypadków śmiertelnych. Wejście nie jest definiowane i nie jest uwzględnianie w analizie. **Kontekstem** jest sytuacja ekonomiczno-polityczna w kraju, np. akty prawne, kryzys ekonomiczny, prowadzone działania wojenne. W pracy [124] po

przeanalizowaniu danych stwierdzono, że zmiany obiektu ze względu na kontekst mają charakter nagły.

W rozpatrywanym okresie, tj. 1851 – 1962, wpływ na obiekt mogą mieć cztery wydarzenia:

- 1887 – wprowadzenie aktu prawnego regulującego zasady technicznego i socjalnego funkcjonowania kopalń w Wielkiej Brytanii (*The Coal Mines Regulations Act*);
- 1914-1918 – I wojna światowa;
- 1929-1933 – wielki kryzys gospodarczy;
- 1939-1945 – II wojna światowa.

Cel badania

Celem badania jest sprawdzenie, czy proponowane w pracy algorytmy wykrywania zmian kontekstu są w stanie wykrywać rzeczywiste momenty zmian. Otrzymane wyniki porównane są z metodami znanymi w literaturze przedmiotu.

Metodyka badania

1. *Przetwarzanie wstępne zbioru danych*
   Oryginalny zbiór danych [108] składa się ze znacznika czasowego (roku) oraz liczby wypadków. Zbiór danych nie zawiera brakujących obserwacji. Liczba wypadków waha się od 0 do 6.

2. *Estymacja rozkładów*
   W literaturze zazwyczaj modeluje się obiekt za pomocą rozkładów Poissona z momentami zmian [1, 32, 45, 55, 92, 108, 124], jednak ze względu na prezentowane algorytmy posłużymy się zmiennymi dyskretnymi o zadanej liczbie Y wartości, $y \in \{0, 1, \ldots, Y-1\}$. Wartość wyjścia odpowiada liczbie wypadków, natomiast ostatnia wartość akumuluje liczby wypadków, które są jej równe lub większe od niej. Do estymacji rozkładów wykorzystano histogramy.

3. *Sposób oceny metod*
   Metody porównano ze względu na wskazane momenty zmian. Przyjmuje się, że jeżeli metoda zwraca momenty pokrywające się z rezultatami podanymi w literaturze przedmiotu lub z czterema wydarzeniami wymienionymi wcześniej, to znaczy, że zmiany kontekstu zostały wykryte poprawnie. Pozostałe wykrycia traktowane są jako błędne.

Uwagi

- Badania, tj. przetwarzanie danych oraz implementacja algorytmu, przeprowadzono w środowisku Matlab®.
- Wyniki dla metod porównanych zaczerpnięto z literatury przedmiotu.

Wyniki

Rezultaty, w zależności od długości okna i wartości parametru wrażliwości, dla algorytmu z modelowaniem częstościowym oraz miarą Bhattacharyya przedstawiono w tabeli 6.1; miarą Kullbacka-Leiblera – w tabeli 6.2; miarą Lina-Wonga – w tabeli 6.3; zmodyfikowaną miarą Lina-Wonga – w tabeli 6.4. Wyniki, w zależności od długości okna oraz liczby wartości wyjścia Y, dla algorytmu z modelowaniem bayesowskim zaprezentowano w tabeli 6.5. W celu porównania otrzymanych wyników z rezultatami znanymi w literaturze przedmiotu, w tabeli 6.6 zebrano wykryte zmiany kontekstów za pomocą metod ze wskazanych źródeł. Na rysunku 6.1 zaznaczono liczbę wypadków podczas jednego roku w kopalniach w Wielkiej Brytanii od 1851 do 1962.

Długość okien L	$\sigma = 0.4$	$\sigma = 0.5$	$\sigma = 0.6$	$\sigma = 0.7$
4	1858, 1863, 1871, 1876, 1884, 1891, 1896, 1906, 1911, 1918, 1928, 1934, 1939, 1945	1862, 1872, 1879, 1884, 1891, 1896, 1908, 1928, 1939	1862, 1872, 1879, 1884, 1891, 1896, 1908, 1928, 1939	1865, 1872, 1879, 1884, 1894, 1910, 1928, 1939
6	1856, 1869, 1877, 1887, 1894, 1904, 1929, 1938	1869, 1878, 1887, 1894, 1904, 1929, 1939	1869, 1878, 1891, 1929, 1939	1869, 1878, 1895, 1929, 1939
8	1868, 1877, 1886, 1895, 1910, 1929, 1938	1886, 1896, 1938	1887, 1896, 1939	1891, 1942
10	1868, 1886, 1929, 1940	1886, 1929, 1940	1886, 1939	**1891, 1939**
12	1886, 1929, 1942	1888, 1929, 1942	1891, 1941	1891, 1942
14	1886, 1940	1887, 1942	1887	1890
16	1886, 1942	1886, 1942	1891	1891
18	1885	1886	1886	-
20	1884	1886	1887	-

Tabela 6.1: Wykryte zmiany kontekstu dla algorytmu z modelowaniem częstościowym oraz miary Bhattacharyya w zależności od długości okien i parametru wrażliwości σ.

Długość okien L	$\sigma = 4.25$	$\sigma = 4.5$	$\sigma = 4.75$	$\sigma = 5$
4	1865, 1872, 1891, 1910, 1932, 1939	1865, 1872, 1891, 1910, 1932, 1939	1865, 1872, 1891, 1910, 1932, 1939	1865, 1872, 1891, 1910, 1932, 1939
6	1891, 1910, 1929, 1939	1891, 1910, 1929, 1939	1891, 1929, 1939	1929, 1939
8	1891, 1932	1891, 1932	1891	-
10	1891, 1929	1891, 1929	1891, 1929	1891
12	1891, 1929	1891, 1929	1891, 1929	1891
14	**1891, 1940**	1891	1891	1891
16	1891, 1942	1891	1891	1891
18	1886	1886	-	-
20	1887	-	-	-

Tabela 6.2: Wykryte zmiany kontekstu dla algorytmu z modelowaniem częstościowym oraz miary Kullbacka-Leiblera w zależności od długości okien i parametru wrażliwości σ.

Długość okien L	$\sigma = 0.2$	$\sigma = 0.25$	$\sigma = 0.3$	$\sigma = 0.35$
4	1858, 1863, 1868, 1875, 1880, 1885, 1891, 1896, 1902, 1907, 1918, 1926, 1932, 1938, 1943, 1956	1862, 1869, 1875, 1880, 1885 1891, 1896, 1904, 1910, 1927, 1932, 1939, 1945, 1956	1862, 1872, 1877, 1884, 1893, 1906, 1911, 1927, 1932, 1939, 1945	1865, 1872, 1879, 1884, 1865, 1872, 1879, 1884, 1894, 1910, 1928, 1939
6	1856, 1869, 1876, 1886, 1893, 1902, 1910, 1926, 1937, 1945	1856, 1869, 1877, 1887, 1894, 1903, 1928, 1938	1856, 1869, 1878, 1887, 1894, 1904, 1929, 1938	1869, 1878, 1895, 1904, 1929 1938
8	1868, 1877, 1886, 1895, 1904, 1926, 1937	1868, 1877, 1886, 1895, 1904, 1926, 1937	1879, 1891, 1937	1887, 1896, 1939
10	1868, 1879, 1890, 1926, 1938	1878, 1889, 1929, 1940	1886, 1939	1887 1939
12	1878, 1891, 1926, 1939	1879, 1892, 1929, 1942	1880, 1893, 1940	**1891, 1941**
14	1879, 1894, 1927, 1942	1879, 1894, 1940	1887, 1940	1890
16	1882, 1926, 1943	1886, 1942	1889, 1942	1891
18	1882	1885	1886	-
20	1884	1884	1886	-

Tabela 6.3: Wykryte zmiany kontekstu dla algorytmu z modelowaniem częstościowym oraz miary Lina-Wonga w zależności od długości okien i parametru wrażliwości σ.

Długość okien L	$\sigma = 0.05$	$\sigma = 0.1$	$\sigma = 0.15$	$\sigma = 0.2$
4	1858, 1863, 1869, 1875, 1880, 1885 1891, 1896, 1904, 1910, 1918, 1927, 1932, 1939, 1945, 1956	1862, 1872, 1879, 1884, 1893 1908, 1928, 1939	1865, 1872, 1879, 1884, 1894, 1910, 1928, 1939	1876, 1884, 1894, 1910, 1928, 1939
6	1856, 1869, 1877, 1886, 1893, 1903, 1910, 1926, 1937, 1945	1856, 1869, 1878, 1887, 1894, 1904, 1929, 1938	1869, 1878, 1895, 1904, 1929, 1939	1929, 1939
8	1868, 1877, 1886, 1895, 1904, 1926, 1937	1886, 1896, 1938	1887, 1896, 1939	1940
10	1868, 1879, 1890, 1926, 1938	1886, 1939	1886, 1939	1939
12	1878, 1891, 1929, 1942	1889, 1941	**1891, 1941**	1891, 1942
14	1879, 1894, 1928, 1943	1887, 1940	1890	-
16	1882, 1940	1890, 1942	1891	-
18	1882	-	-	-
20	1884	-	-	-

Tabela 6.4: Wykryte zmiany kontekstów dla algorytmu z modelowaniem częstościowym oraz zmodyfikowanej miary Lina-Wonga w zależności od długości okien i parametru wrażliwości σ.

Długość okna L	$Y = 2$	$Y = 3$	$Y = 4$	$Y = 5$
10	1911, 1929, 1939	1911, 1939	-	-
12	1896, 1904, 1929, 1939	1929, 1939	1939	-
14	1891, 1910, 1929, 1939	1929, 1940	-	-
16	1887, 1896, 1910, 1929, 1939	1891, 1940	1942	-
18	1886, 1896, 1911, 1929, 1939	1886, 1940	1891, 1939	-
20	1886, 1926, 1939	1887, 1929, 1940	1887, 1939	**1891, 1940**
22	1887, 1929, 1941	1887, 1929, 1941	1887, 1929, 1941	1891, 1942
24	1887, 1926, 1941	1889, 1929, 1942	1889, 1929, 1942	1891, 1942
26	1886, 1926, 1940	1880, 1894, 1929, 1943	1886, 1929, 1943	1890, 1942
28	1886, 1926, 1941	1880, 1928, 1943	1886, 1940	1887, 1942
30	1885, 1926, 1942	1880, 1896, 1927, 1943	1886, 1940	1889, 1942
32	1885, 1926, 1943	1886, 1926, 1943	1886, 1942	1887, 1942
34	1884, 1941	1882, 1942	1882, 1942	1886
36	1883, 1929	1882, 1942	1883, 1942	1886
38	1883, 1942	1883, 1942	1883	1885
40	1884, 1941	1875, 1896, 1942	1884	1884

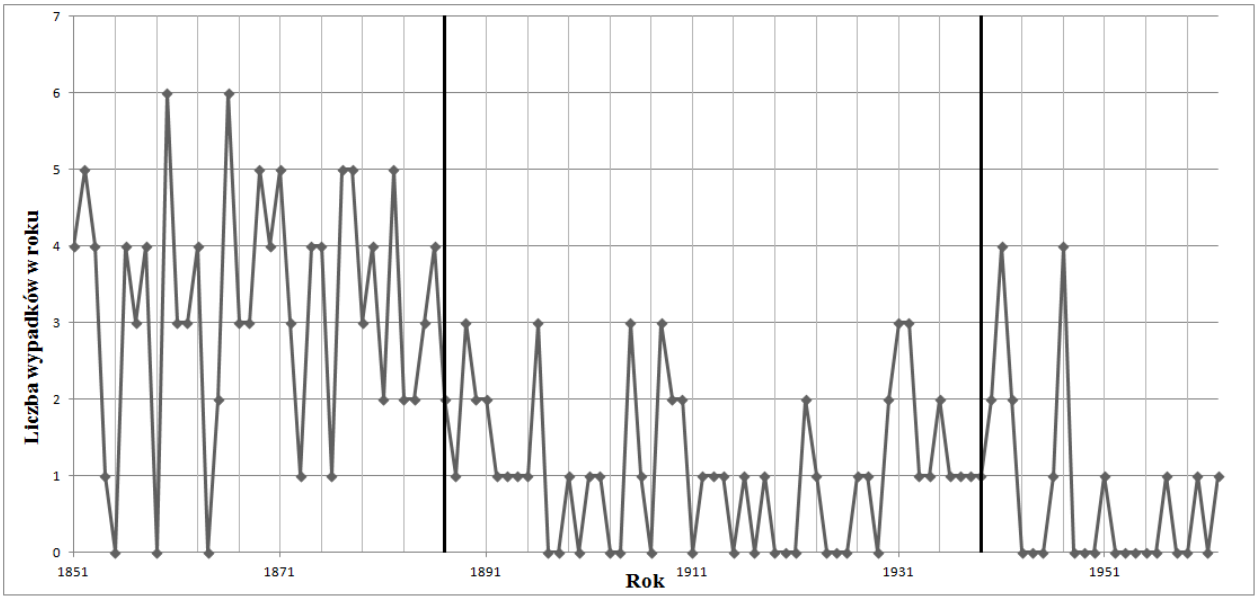
Tabela 6.5: Wykryte zmiany kontekstu dla algorytmu z modelowaniem bayesowskim dla $\sigma = 2$ w zależności od długości okna oraz liczby wartości wyjścia Y.

Lp.	Źródło	Sposób analizy	Wykryte zmiany kontekstu
1.	[1]	prospektywny	1895
2.	[32]	retrospektywny	1891, 1940
3.	[55]	retrospektywny	1890, 1940
4.	[45]	retrospektywny	1890, 1940
5.	[92]	retrospektywny	1890
6.	[108]	prospektywny	1891
7.	[124]	retrospektywny	1890

Tabela 6.6: Wykryte zmiany kontekstu dla metod znanych w literaturze przedmiotu.

Miara	Wartość L	Wartość σ	Wykryte zmiany kontekstu
Bhattacharyya	10	0.7	1891, 1939
Kullback-Leibler	14	4.25	1891, 1940
Lin-Wong	12	0.35	1891, 1941
zmod. Lin-Wong	12	0.15	1891, 1941
Podejście bayesowskie ($Y = 5$)	20	2	1891, 1940

Tabela 6.7: Wykryte zmiany kontekstu dla proponowanych w niniejszej pracy algorytmów dla wybranych wartości długości okna i parametru wrażliwości.



Rysunek 6.1: Liczba wypadków podczas jednego roku w kopalniach w Wielkiej Brytanii od 1851 do 1962. Pogrubionymi, pionowymi liniami zaznaczono wprowadzenie aktu prawnego regulującego zasady funkcjonowania kopalń oraz początek II wojny światowej.

Dyskusja

Korzystając z wyników otrzymanych za pomocą metod znanych w literaturze przedmiotu (patrz tabela 6.6) można przyjąć, że w ciągu uczącym zachodzą dwie zmiany kontekstu – ok. 1890 i 1939. Pierwsza data związana jest z wprowadzeniem aktu prawnego regulującego sposób funkcjonowania kopalń w 1887, natomiast druga – z początkiem II wojny światowej.

Zarówno w podejściu częstościowym i bayesowskim wykryto oba momenty zmiany (patrz tabela 6.7), zatem można przyjąć, że algorytmy są w stanie wykryć rzeczywiste zmiany.

W podejściu częstościowym dobór długości okien oraz wartości parametru wrażliwości wpływa na liczbę wykrywanych zmian. Błędne określenie obu wielkości skutkuje wykrywaniem zbyt dużej lub zbyt małej liczby zmian. Podobne wnioski płyną dla podejścia bayesowskiego, w którym istotne jest określenie długości okna oraz liczby wartości wyjścia. Jednak w podejściu bayesowskim kluczowe pozostałe określenie długości okna, ponieważ liczba wartości wyjścia w zadaniu ekstrakcji wiedzy zazwyczaj jest znana z góry.

Analizując proponowane w pracy metody pod kątem liczby i poprawności wykrywania zmian kontekstu można wyciągnąć następujące wnioski. Modyfikacja miary Lina-Wonga powoduje zmniejszenie liczby wykrytych zmian w porównaniu z miarą Lina-Wonga. Tym niemniej obie miary, jak również miara Bhattacharyya, dla błędnie dobranych wartości długości okna oraz parametru wrażliwości, zwracają wiele błędnych momentów zmian. W podejściu bayesowskim istotne jest odpowiednie dobranie długości okna. Ze względu na przyjęte założenia, dla zbyt dużego okna, część zmian nigdy nie zostanie wykrytych. Wynika to z faktu porównywania dwóch modeli – bez zmiany i z jedną zmianą. Jeżeli obserwacje zawarte w oknie przesuwnym uwzględniają dwie zmiany, to jedna nie zostanie wykryta. Podobna sytuacja zachodzi, gdy długość okna jest zbyt mała, ponieważ model bez zmian będzie najczęściej preferowany.

Cel badania został osiągnięty. W podejściu częstościowym i bayesowskim, dla odpowiednio dobranych wartości parametrów, uzyskano te same rezultaty jak w przypadku metod znanych w literaturze przedmiotu.

6.3 Zadanie wykrywania zmian w zastosowaniu do systemów zorientowanych na usługi

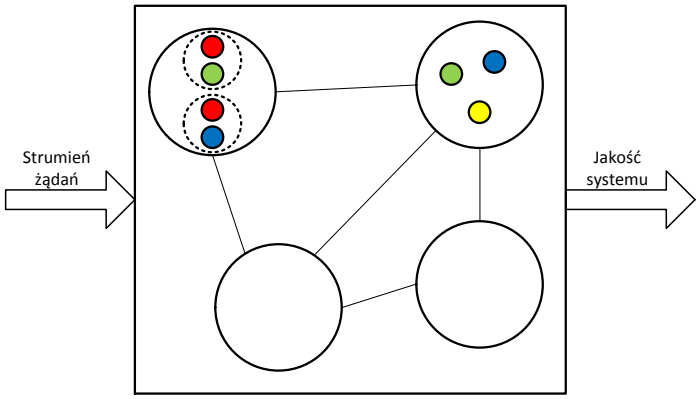
Opis problemu

W ostatnich latach wzrosło zainteresowanie systemami zorientowanymi na usługi (ang. *service-oriented systems*, SoS), które funkcjonują zgodnie z paradygmatem SOA [43]. Ogólnie mówiąc, systemy tego typu są systemami komputerowymi o rozproszonej strukturze (ang. *distributed computer systems*), w których aplikacje nazywane są *usługami złożonymi* (ang. *complex* lub *composite service*). Każda usługa złożona składa się z usług atomowych (ang. *atomic service*), które dostarczają wymaganych funkcjonalności na określonym poziomie jakości wykonania usługi (ang. *Quality-of-Service*, QoS) [101]. Usługi mogą dostarczać tych samych funkcjonalności, ale są rozmieszczone w różnych lokalizacjach w sieci w ramach węzłów obliczeniowych. Węzły obliczeniowe składają się na system wykonawczy, którego celem jest spełnienie żądań użytkowników dotyczących wybranych funkcjonalności.

Formalnie system wykonawczy, który jest rozpatrywanym **obiektem niestacjonarnym**, można zdefiniować jako graf $\mathcal{G}_W = (\mathcal{V}_W, \mathcal{E}_W)$, w którym $\mathcal{V}_W$ oznacza zbiór węzłów obliczeniowych, natomiast $\mathcal{E}_W$ – zbiór kanałów komunikacyjnych pomiędzy węzłami obliczeniowymi. Zakładamy, że wyróżnia się S usług dostępnych w SoS, których wersje mogą znajdować się na różnych węzłach obliczeniowych. Ponadto każdemu węzłowi obliczeniowemu przydzielone są zasoby obliczeniowe, np. liczba procesorów. **Wejściem** do systemu wykonawczego jest strumień żądań wykonania usług, natomiast **wyjściem** – jakość systemu wyrażona za pomocą wybranych wskaźników jakości. W niniejszym punkcie rozpatrzymy średnie opóźnienie wykonania usługi (ang. *latency*), tj. czas trwania między wejściem żądania do systemu wykonawczego a ostatecznym wykonaniem usługi. **Kontekstem** dla systemu wykonawczego może być zmiana strumienia żądań usług, np. w zależności od pory dnia lub istniejących zapotrzebowań, albo zmiana w działaniu samego systemu wykonawczego, np. z powodu awarii lub modernizacji systemu. W obu przypadkach zmiana może wpływać na jakość systemu.

Przykładowy system wykonawczy z zaznaczeniem wejścia i wyjścia przedstawiono na

rysunku 6.2. Węzły obliczeniowe oznaczono czarnymi okręgami. Przerywane okręgi określają maszyny wirtualne działające w ramach węzła obliczeniowego. Usługi atomowe oznaczono kolorowymi kółkami. Połączenia między węzłami obliczeniowymi określają istniejące kanały komunikacyjne.



Rysunek 6.2: Przykładowy system wykonawczy (obiekt niestacjonarny) w systemie zorientowanym na usługi. Połączone okręgi oznaczają węzły obliczeniowe wraz z kanałami komunikacyjnymi. Okręgi przerywane reprezentują maszyny wirtualne, na których działają usługi atomowe lub złożone (kolorowe kółka). Wejściem obiektu jest strumień żądań, natomiast wyjściem – jakość systemu.

Dla tak sformułowanego systemu wykonawczego można stawiać różne zadania decyzyjne. Jednym z zagadnień jest realokacja zasobów obliczeniowych między węzłami obliczeniowymi w celu utrzymania jakości systemu przy zmianach kontekstu, np. zmiennych strumieniach żądań [130]. Innym zadaniem jest analiza zmian wyjścia systemu wykonawczego w celu ekstrakcji wiedzy o działaniu systemu ze względu na różne pory dnia. Dodatkowo można badać zmiany w jakości systemu dla wykrywania anomalii w strumieniu żądań lub w działaniu samego systemu wykonawczego, np. awarii. W każdym z wymienionych przypadków rozsądnym wydaje się być stosowanie metod wykrywania zmian kontekstu, dzięki którym można przeprowadzić analizę obiektu dla różnych kontekstów. Dalej skupimy się wyłącznie na zagadnieniu wykrywania zmian kontekstu bez określania konkretnego zadania podejmowania decyzji, dla którego wykrywanie jest przeprowadzane. Dodatko-

wo warto zauważyć, że wykrywanie zmian można przeprowadzać na podstawie zarówno wejścia i wyjścia. Jednak w procesie podejmowania decyzji zazwyczaj interesuje nas utrzymanie wyjścia obiektu na określonym poziomie (regulacja), dlatego też zmiany kontekstu wykrywane będą wyłącznie w oparciu o wyjście obiektu.

Cel badania

Celem badania jest sprawdzenie, czy stosowanie proponowanych algorytmów wykrywania zmian kontekstu z modelowaniem częstościowym oraz bayesowskim pozwala na wykrycie zmian kontekstu w systemie wykonawczym, w którym mierzoną wielkością jest wybrany wskaźnik jakości, tj. średnie opóźnienie wykonania żądania.

Metodyka badania

Przyjęto następującą metodykę działania:

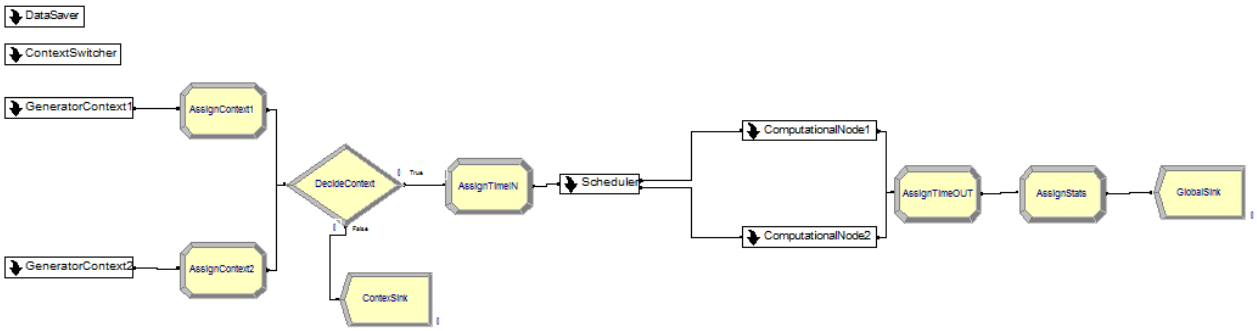
1. *Przygotowanie środowiska symulacyjnego*
   Ze względu na złożoność systemów usługowych, realizacja założonego celu przeprowadzona była w drodze badań symulacyjnych. Generator żądań oraz system wykonawczy został zaimplementowany w środowisku symulacji zdarzeń dyskretnych *Arena* [4]. Przyjęto następujące założenia:

   - rozpatrzono 2 węzły obliczeniowe;
   - w ramach każdego węzła obliczeniowego znajdowały się 2 maszyny wirtualne;
   - zasoby obliczeniowe w liczbie 16 procesorów rozlokowano po równo na każdy węzeł;
   - łączna liczba usług złożonych wynosiła 4;
   - każdy węzeł obliczeniowy oraz każda maszyna wirtualna zamodelowana została jako kolejka FIFO z procesorem.

   Symulator systemu wykonawczego składa się z następujących elementów (patrz rysunek 6.3):

   - generatora żądań dla każdej usługi (*GeneratorContext1* i *GeneratorContext2* na rysunku 6.3);

- elementu rozdzielającego żądania usług do węzłów obliczeniowych (*Scheduler* na rysunku 6.3), który przekazuje żądanie do usługi, która posiada najmniejszą liczbę oczekujących żądań;
- węzłów obliczeniowych (*ComputationalNode1* i *ComputationalNode2* na rysunku 6.3); każdy węzeł obliczeniowy zawiera 2 maszyny wirtualne z przydzielonymi zasobami obliczeniowymi, tj. liczbą procesorów.



Rysunek 6.3: Symulator systemu wykonawczego w środowisku *Arena*.

2. *Modelowanie generatora strumienia żądań wykonania usług*
   W klasycznej teorii obsługi masowej żądania do systemów usługowych, które dotyczą np. nawiązania połączenia telekomunikacyjnego, modelowane są za pomocą procesów Poissona [54]. Jednak w rozległych systemach teleinformatycznych zaobserwowano, że procesy Poissona można wykorzystywać jedynie do modelowania strumieni otwierania nowych sesji użytkownika, natomiast modele takie zawodzą w przypadku generowania pakietów TCP/IP [119].

   W rozpatrywanym przypadku systemu zgodnego z paradygmatem SOA interesuje nas modelowanie strumieni żądań wykonania usług. Zakładamy, że klienci niezależnie od siebie zgłaszają żądania o stałych wielkościach. Następnie przyjmujemy, że całe żądanie wysyłane jest do systemu wykonawczego, w którym jest przetwarzane, a następnie wykonywane. W związku z takimi założeniami strumienie żądań wykonania usług można traktować identycznie jak strumienie otwierania nowych sesji użytkownika. Dlatego w środowisku symulacyjnym żądania wykonania usług modelowane są

za pomocą procesów Poissona o zadanej intensywności.

3. *Modelowanie działania węzłów obliczeniowych*
   W [101] szczegółowo omówiono architekturę rozproszonego systemu komputerowego składającego się z wielu serwerów. Schemat systemu wykonawczego oparto więc na strukturze prezentowanej w [101], w której serwer WWW (ang. *web server*) można utożsamiać z węzłem obliczeniowym.

   Wartości parametrów przetwarzania żądań na węzłach obliczeniowych ustalono na podstawie badań *benchmarkowych* przeprowadzonych w [95]. We wskazanej pracy zaprezentowano kilka testów, w których porównywano działanie serwerów WWW, m.in. Apache, LiteSpeed. W celu przeprowadzenia niniejszego eksperymentu skorzystano z testu *Small Static File (KeepAlive)* dla jednego serwera WWW, które polegało na ciągłym wysyłaniu pliku o zadanym rozmiarze (100B) do serwera i sprawdzaniu, czy połączenie istnieje. W teście tym średnia liczba żądań przetwarzanych przez serwer na 1 sekundę wynosiła ok. 2500, czyli 0.0004 sekundy na 1 żądanie. Dodatkowo przyjęto, że czas przetwarzania na maszynie wirtualnej wynosi 0.0008 sekundy.

   Następnie ustalono, że każdy węzeł obliczeniowy posiada zasoby obliczeniowe w liczbie 8 procesorów. Ponadto przyjęto, że:

   - pierwszej maszynie wirtualnej na pierwszym węźle obliczeniowym przydzielono 6 procesorów;
   - drugiej maszynie wirtualnej na pierwszym węźle obliczeniowym przydzielono 2 procesory;
   - pierwszej i drugiej maszynie wirtualnej na drugim węźle obliczeniowym przydzielono po 4 procesory.

4. *Modelowanie działania usług*
   W badaniu przyjęto, że żądania dotyczą wykonania 4 usług rozlokowanych na węzłach obliczeniowych. Wykorzystano rzeczywiste usługi rozwiązujące zadanie eksploracji danych, które zaimplementowano w ramach systemu *Service Oriented Data Mining* w języku Java i z zastosowaniem biblioteki WEKA [122]. Wybrano usługi rozwiązujące zadanie klasyfikacji przy użyciu:

- wielowarstwowej sieci perceptronów – *Multilayer Perceptron*;
- regresji logistycznej – *Logistic Regression*;
- drzewa decyzyjnego – *J48*;
- modelu probabilistycznego z założeniem niezależności stochastycznej wejść – *Naïve Bayes*.

W celu ustalenia czasu przetwarzania pojedynczego żądania[1] o ustalonej wielkości (przyjęto 16kB) przez każdą usługę wykorzystano środowisko testowe *soapUI* [138]. Otrzymano następujące wartości:

- dla *Multilayer Perceptron*: czas minimalny $= 0.051$[s], czas maksymalny = 0.672[s], czas średni $= 0.088$[s];
- dla *Logistic Regression*: czas minimalny $= 0.028$[s], czas maksymalny = 0.214[s], czas średni $= 0.045$[s];
- dla *J48*: : czas minimalny $= 0.005$[s], czas maksymalny = 0.183[s], czas średni $= 0.011$[s];
- dla *Naïve Bayes*: : czas minimalny $= 0.006$[s], czas maksymalny = 0.053[s], czas średni $= 0.01$[s].

Czas przetwarzania pojedynczego żądania jest modelowany za pomocą rozkładu trójkątnego o zadanych wartościach minimalnej, maksymalnej i średniej. Dodatkowo przyjęto, że czas przetwarzania żądania jest dzielony przez liczbę procesorów przydzielonych maszynie wirtualnej, na której usługa się znajduje.

Ze względu na średni czas przetwarzania żądań usługi rozmieszczono w następujący sposób (w nawiasach podano łączną liczbę przypadających procesorów na usługę):

- *Multilayer Perceptron*: maszyna wirtualna nr 1 na węźle obliczeniowym nr 1 oraz maszyna wirtualna nr 1 na węźle obliczeniowym nr 2 (łącznie 5 procesorów);
- *Logistic Regression*: maszyna wirtualna nr 1 na węźle obliczeniowym nr 1 oraz maszyna wirtualna nr 1 na węźle obliczeniowym nr 2 (łącznie 5 procesorów);

[1] Żądanie dotyczy budowy klasyfikatora.

- *J48*: maszyna wirtualna nr 2 na węźle obliczeniowym nr 1 oraz maszyna wirtualna nr 2 na węźle obliczeniowym nr 2 (łącznie 3 procesory);
- *Naïve Bayes*: maszyna wirtualna nr 2 na węźle obliczeniowym nr 2 (łącznie 2 procesory).

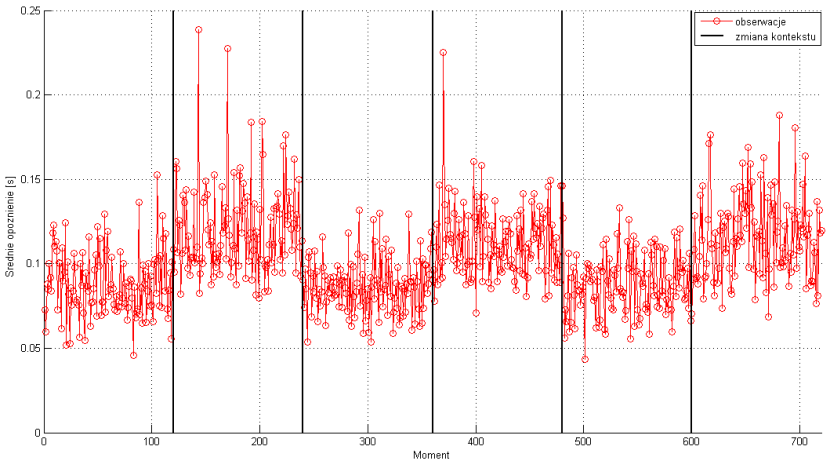
5. *Przypadki użycia*
   W niniejszym badaniu rozpatrzono następujące przypadki:

   (a) Zmiana kontekstu dotyczy strumienia żądań, w którym intensywności procesów Poissona zwiększają się trzykrotnie. Przyjęto, że wartości intensywności strumieni żądań usług *Multilayer Perceptron*, *Logistic Regression* i *J48* wynoszą $1/3$, natomiast *Naïve Bayes* – 1 przed zmianą i, odpowiednio, 1 i 3 po zmianie kontekstu. Założono, że na horyzoncie 7200 obserwacji nastąpiło 5 zmian kontekstu (przełączeń między wartościami intensywności, patrz rysunek 6.4).

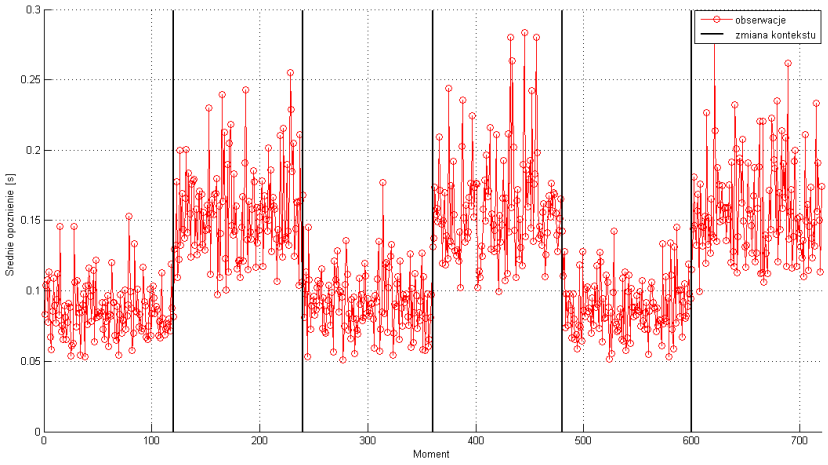
   (b) Zmiana kontekstu dotyczy strumienia żądań, w którym intensywności procesów Poissona zwiększają się sześciokrotnie. Przyjęto, że wartości intensywności strumieni żądań usług *Multilayer Perceptron*, *Logistic Regression* i *J48* wynoszą $1/3$, natomiast *Naïve Bayes* – 1 przed zmianą i, odpowiednio, 2 i 6 po zmianie kontekstu. Założono, że na horyzoncie 7200 obserwacji nastąpiło 5 zmian kontekstu (przełączeń między wartościami intensywności, patrz rysunek 6.5).

   (c) Zmiana kontekstu dotyczy awarii maszyny wirtualnej nr 1 na węźle obliczeniowym nr 1, co skutkuje zmianą zasobów obliczeniowych (zamiast 6 procesorów dostępne są 2). Przyjęto, że na całym horyzoncie obserwacji wartości intensywności strumieni żądań usług *Multilayer Perceptron*, *Logistic Regression* i *J48* wynoszą 1, natomiast *Naïve Bayes* – 2. Założono, że na horyzoncie 3600 obserwacji nastąpiły 2 zmiany kontekstu (zmniejszenie liczby procesorów i przywrócenie pierwotnej liczby, patrz rysunek 6.6).

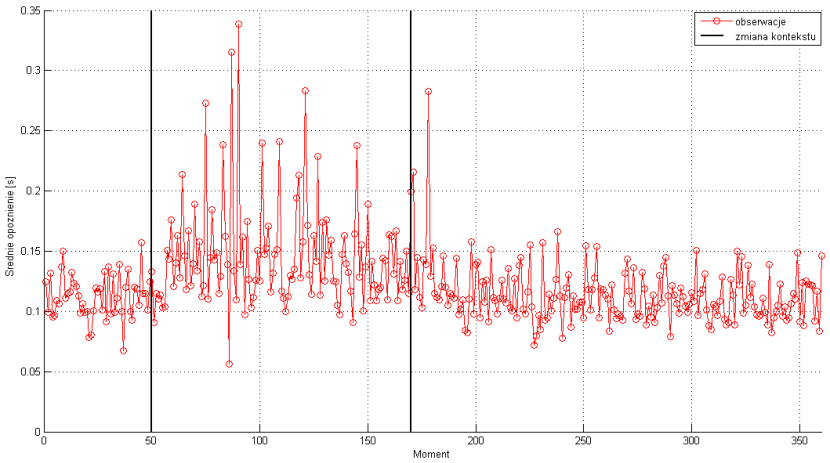
   Pierwsze dwa przypadki obrazują sytuację, gdy zmiana kontekstu wpływa na obiekt, tj. system wykonawczy, pośrednio poprzez zmianę strumienia żądań wykonania usług. Natomiast trzeci przypadek przedstawia zmianę kontekstu, która bezpośrednio wpływa na działanie obiektu.



Rysunek 6.4: Przykładowe obserwacje średniego opóźnienia wykonania usługi w systemie wykonawczym (kolor czerwony) z zaznaczonymi momentami zmian kontekstu (czarne pionowe linie) dla przypadku (a).



Rysunek 6.5: Przykładowe obserwacje średniego opóźnienia wykonania usługi w systemie wykonawczym (kolor czerwony) z zaznaczonymi momentami zmian kontekstu (czarne pionowe linie) dla przypadku (b).



Rysunek 6.6: Przykładowe obserwacje średniego opóźnienia wykonania usługi w systemie wykonawczym (kolor czerwony) z zaznaczonymi momentami zmian kontekstu (czarne pionowe linie) dla przypadku (c).

6. *Estymacja rozkładów*
   Rozkłady estymowane były za pomocą histogramów dla następujących przedziałów:
   - dla przypadku (a): od 0 do 0.2 z krokiem 0.05;
   - dla przypadku (b): od 0 do 0.25 z krokiem 0.025;
   - dla przypadku (c): od 0 do 0.2 z krokiem 0.05.
7. *Sposób oceny metod*
   W każdym przypadku znana jest liczba zmian kontekstu. W związku z tym metody i różne miary porównano przy pomocy liczby poprawnie i niepoprawnie wykrytych zmian kontekstu.

Uwagi

- Średnie opóźnienie wykonania usługi było wyznaczane na podstawie obserwacji dla ostatnich 10 jednostek czasu.

- Zmiany wskazywane w momencie, który różnił się od rzeczywistego momentu zmiany kontekstu o więcej niż 10 jednostek czasu, były uznawane za niepoprawne.
- Dla każdego przypadku uruchomiono symulator 10 razy. Wszystkie uzyskane wyniki uśredniono.
- Wartości parametru wrażliwości i długości okien dla podejścia częstościowego z wykorzystaniem miar Bhattacharyya, Kullbacka-Leiblera, Lina-Wonga i zmodyfikowanej miary Lina-Wonga, oraz dla podejścia bayesowskiego przedstawiono w tabelach 6.8, 6.9, 6.10. Podane wartości parametrów uzyskano po kilkukrotnym uruchomieniu algorytmów na ustalonym ciągu uczącym.

Wyniki

Uzyskane wyniki dla przypadku z trzykrotnym zwiększeniem intensywności strumieni żądań wykonania usług przedstawiono w tabeli 6.8. Rezultaty dla przypadku z sześciokrotnym zwiększeniem intensywności strumieni żądań wykonania usług zebrano w tabeli 6.9. Natomiast liczbę poprawnie i niepoprawnie wykrytych zmian dla przypadku z awarią jednej z maszyn wirtualnych zaprezentowano w tabeli 6.10.

Dyskusja

Analizując otrzymane wyniki można ogólnie stwierdzić, że większość zmian kontekstu w rozpatrywanych przypadkach została poprawnie wykryta przez metody z podejściem częstościowym i bayesowskim. Najmniej zmian zostało wykrytych w przypadku z trzykrotną zmianą wartości intensywności strumienia żądań oraz w przypadku awarii. Jednak rezultaty te znajdują uzasadnienie, jeżeli przeanalizuje się obserwacje wyjścia obiektu (patrz rysunek 6.4 i 6.6). Dla przypadku ze zmianą intensywności część zmian kontekstu powodowało stosunkowo niewielkie różnice wartości wyjścia, przez co zmiana nie była zgłaszana. Sytuacja taka jest szczególnie widoczna przy drugiej zmianie w przypadku awarii, gdzie system wykonawczy po chwilowym „zaburzeniu”, tj. zwiększeniu długości kolejek na maszynach wirtualnych, zaczyna stabilizować swoją pracę i średnie opóźnienie maleje

niezależnie od przywrócenia pierwotnej liczby procesorów. Dlatego druga zmiana nie była zazwyczaj zgłaszana.

Najprostsza do analizy sytuacja była dla sześciokrotnej zmiany intensywności strumienia żądań wykonania usług. W tym przypadku metody umożliwiły wykrycie prawie wszystkich zmian.

Natomiast porównując podejście częstościowe i bayesowskie można wysnuć przypuszczenie, że modelowanie bayesowskie pozwala na uzyskanie nieznacznie lepszych rezultatów. Widać to szczególnie w przypadku (b) (patrz tabela 6.9) i (c) (patrz tabela 6.10). Ponadto metoda z modelowaniem bayesowskim we wszystkich przypadkach zwracała najmniej niepoprawnie wykrytych zmian, co w zastosowaniu do systemów zorientowanych na usługi może mieć istotne znaczenie, np. uruchomienie procedury realokacji zasobów wiąże się z dodatkowym narzutem obliczeniowym i co za tym idzie – dodatkowym kosztem.

Dodatkowo, korzystając z otrzymanych wyników, wydaje się, że w podejściu częstościowym miara Bhattacharyya najlepiej nadaje się do stosowania do wykrywania zmian kontekstu. We wszystkich przypadkach (oprócz poprawnej liczby wykrytych zmian kontekstu dla miary Kullbacka-Leiblera w przypadku (b)) stosowanie tej miary prowadziło do uzyskania największej liczby wykrytych zmian i najmniejszej liczby niepoprawnie wykrytych.

Jednak dla wszystkich miar niepodobieństwa, również miary Bhattacharyya, pozostaje problem wyboru wartości parametru wrażliwości oraz długości okna przesuwnego. Pod tym względem preferowane jest podejście bayesowskie, w którym należy określić długość okna, zaś wartość parametru wrażliwości można przyjąć wg interpretacji Jeffreysa (patrz tabela 3.1).

Cel badania został osiągnięty. Metody z podejściem częstościowym i bayesowskim pozwalają na poprawne wykrywanie zmian kontekstu. Rozpatrywane trzy przypadki pokazały, że proponowane rozwiązania są skuteczne w sytuacjach z nagłymi zmianami, natomiast zawodzą dla zmian ciągłych.

Miara	**Poprawnie wykryte (max. 5)**	**Niepoprawnie wykryte**
Bhattacharyya ($L = 25, \sigma = 0.2$)	3.2	0.2
Kullback-Leibler ($L = 25, \sigma = 1$)	3.8	0.8
Lin-Wong ($L = 25, \sigma = 0.15$)	2.8	0.7
zmod. Lin-Wong ($L = 25, \sigma = 0.02$)	2.9	0.9
Podejście bayesowskie ($L = 50, \sigma = 0$)	3	0.2

Tabela 6.8: Średnia liczba poprawnie i niepoprawnie wykrytych zmian kontekstu dla przeprowadzonego eksperymentu w przypadku (a).

Miara	**Poprawnie wykryte (max. 5)**	**Niepoprawnie wykryte**
Bhattacharyya ($L = 25, \sigma = 0.5$)	4.6	0.1
Kullback-Leibler ($L = 25, \sigma = 2$)	4.8	0.2
Lin-Wong ($L = 20, \sigma = 0.3$)	4.6	0.3
zmod. Lin-Wong ($L = 20, \sigma = 0.1$)	4.6	0.2
Podejście bayesowskie ($L = 40, \sigma = 2$)	5	0

Tabela 6.9: Średnia liczba poprawnie i niepoprawnie wykrytych zmian kontekstu dla przeprowadzonego eksperymentu w przypadku (b).

Miara	**Poprawnie wykryte (max. 5)**	**Niepoprawnie wykryte**
Bhattacharyya ($L = 60, \sigma = 0.15$)	1	0.3
Kullback-Leibler ($L = 50, \sigma = 1$)	0.7	0.3
Lin-Wong ($L = 40, \sigma = 0.18$)	1.1	0.1
zmod. Lin-Wong ($L = 40, \sigma = 0.035$)	1	0.1
Podejście bayesowskie ($L = 50, \sigma = 2$)	1.1	0.1

Tabela 6.10: Średnia liczba poprawnie i niepoprawnie wykrytych zmian kontekstu dla przeprowadzonego eksperymentu w przypadku (c).

6.4 Zadanie ekstrakcji wiedzy w przypadku deterministycznym – *STAGGER*

Opis problemu

Zbiór danych *STAGGER* jest podstawowym i jednym z pierwszych zbiorów wykorzystywanych do wstępnej oceny metod ekstrakcji wiedzy z uczeniem przyrostowym [134]. Zakłada się obiekt deterministyczny, w którym wyszczególnia się trzy wejścia:

- u^1 – kolor (niebieski, czerwony, zielony; $K_1 = 3$),
- u^2 – kształt (kulisty, trójkątny, prostokątny; $K_2 = 3$),
- u^3 – rozmiar (mały, średni, duży; $K_3 = 3$),

oraz jednego wyjście $y \in \{0, 1\}$.

Przyjmuje się istnienie **kontekstu**, który wpływa na zmianę obiektu w dwóch momentach – 41 i 81. Wówczas obiekt na różnych przedziałach czasowych opisany jest odmienną własnością, tj.

- od momentu 1 do 40 obiekt opisany jest własnością $\phi = (\alpha^1_{czerwony} \wedge \alpha^3_{maly} \Rightarrow \alpha^{out}_1)$, zaś klasa 0 dla wszystkich pozostałych wyrażeń,
- od kroku 41 do 80 obiekt zmienia swoją własność i jest opisany $\phi_1 = (\alpha^1_{zielony} \Rightarrow \alpha^{out}_1)$ lub $\phi_2 = (\alpha^2_{kulisty} \Rightarrow \alpha^{out}_1)$ i klasa 0 dla pozostałych wyrażeń,
- od momentu 81 do 120 własność obiektu zmienia się na $\phi_1 = (\alpha^3_{sredni} \Rightarrow \alpha^{out}_1)$ lub $\phi_2 = (\alpha^3_{duzy} \Rightarrow \alpha^{out}_1)$ i klasa 0 dla pozostałych wyrażeń.

Cel badania

Celem badania jest porównanie ze względu na kryterium (1.7) poprawności działania algorytmów ekstrakcji wiedzy regułowej prezentowanych w niniejszej pracy, tj. AQ-P1, AQ-P2, GRI, z metodami znanymi w literaturze przedmiotu, tj. AQ-PM, FLORA2, dla deterministycznego obiektu niestacjonarnego.

Metodyka badania

1. *Przygotowanie środowiska symulacyjnego*
   Wartość każdego wejścia generowana była wg rozkładu jednostajnego, natomiast wartość wyjścia określana była wg własności odpowiadającej momentowi symulacji. Czas trwania symulacji wynosił 120 momentów.

2. *Ustalenie parametrów algorytmów*
   W celu poprawności działania algorytmów AQ-P1 oraz AQ-P2 wykorzystano własność wynikającą z twierdzenia 2.1, tzn. aby uzyskać dobrą jakość wiedzy w sensie średnim należy dysponować ciągiem uczącym o rozmiarze $N >$*VD-dim*, który w tym przypadku wynosi $N > 27$ (bo $K_1 \cdot K_2 \cdot K_3 = 27$).

   Ponadto, po kilku uruchomieniach każdego algorytmy, wybrano najlepsze parametry, tzn. dla AQ-P1 długość okna ustalono na $L = 30$ (korzystając z podanego twierdzenia) i liczbę różnic na $C_1 = 1$; dla AQ-P2 długość okna – $L = 30$, liczba obserwacji do policzenia funkcji oceniającej $C_2 = 11$, wartość progowa $C_3 = 0.8$; dla GRI wybrano współczynnik zapominania $\gamma = 0.9$, natomiast $\beta = 0.3$.

3. *Przeprowadzenie eksperymentu*
   Przyjęto, że dane napływały w strumieniu i dodatkowo dla każdego kontekstu dysponowano osobnym zbiorem testowym. Wygenerowano 10 ciągów treningowych (120 obserwacji każdy) oraz po 100 obserwacji testowych na jeden kontekst (łącznie 300 obserwacji na jedno uruchomienie symulacyjne).

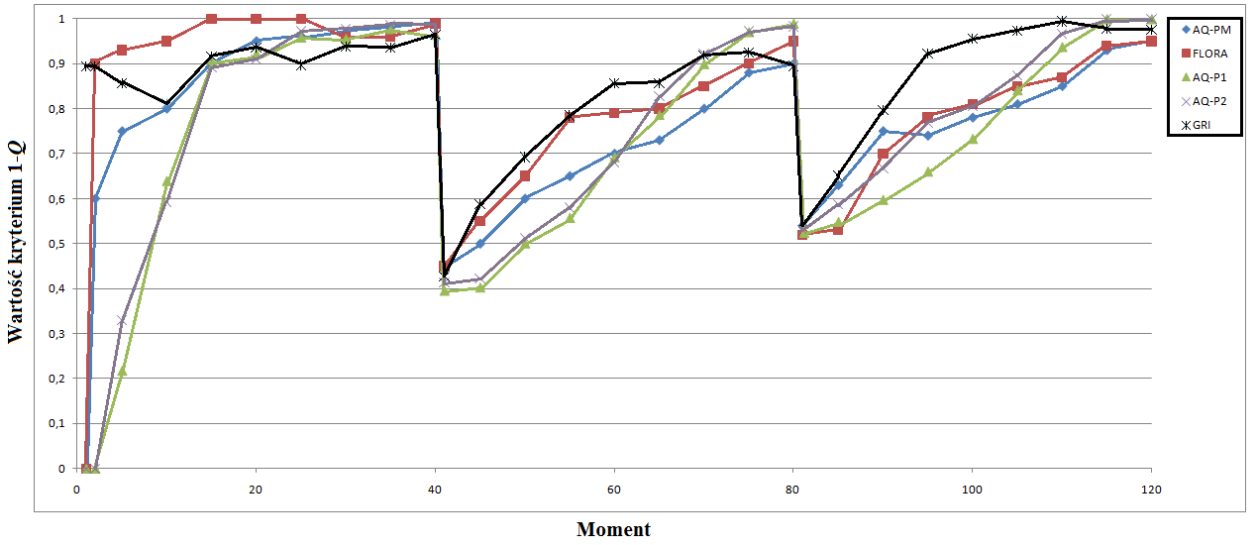
4. *Sposób oceny metod*
   Metody oceniano wg kryterium $1 - Q$, gdzie Q jest zdefiniowane jak (1.7).

Uwagi

- Generator danych oraz algorytmy zaimplementowano w środowisku Matlab®.
- Rezultaty dla AQ-PM zaczerpnięto z pracy [98], zaś dla FLORA2 z [163].

Wyniki

Na rysunku 6.7 przedstawiono wyniki dla rozpatrywanych metod oraz dla metod porównawczych. Kolorem niebieskim oznaczono metodę AQ-PM, kolorem czerwonyn – FLORA2, natomiast kolorem zielonym – AQ-P1, fioletowym – AQ-P2, zaś czarnym – GRI.



Rysunek 6.7: Wyniki rozpatrywanych metod dla zbioru danych *STAGGER*.

Dyskusja

Algorytmy AQ-P1 i AQ-P2 dały zbliżone wyniki i osiągnęły wartość kryterium $1 - Q$ powyżej wartości 0.9 odpowiednio dla każdego kontekstu po ok. 20, 30 i 25 momentach. Natomiast w przypadku algorytmu GRI poziom ten został otrzymany po ok. 20, 30 i 15 momentach. Dla porównania algorytm AQ-PM uzyskał wskazaną wartość kryterium po ok. 20, 40 i 35 momentach, natomiast FLORA2 – po ok. 5, 35 i 35 momentach. Zatem można uznać, że proponowane w niniejszej pracy algorytmy pozwalają na skuteczne „poznanie" obiektu deterministycznego oraz stosowane mechanizmy zapominania umożliwiają ponowne wyuczanie się własności obiektu po nastąpieniu zmiany. Dodatkowo należy zaznaczyć, że algorytmy AQ-P1 i AQ-P2 działały nieznacznie lepiej od metod porównawczych, natomiast algorytm GRI osiągnął najlepsze rezultaty (w przypadku pierwszego kon-

tekstu od 5. momentu uzyskał wartość kryterium pow. 0.9, która później przez 15 momentów nieznacznie spadła do poziomu 0.8).

Wszystkie uzyskane wyniki są uśrednione z dziesięciu uruchomień, w których ciągi uczące zawierały obserwacje w różnej kolejności. W związku z tym prezentowane w pracy algorytmy ekstrahują poprawną wiedzę ze względu na podejmowanie decyzji niezależnie od kolejności obserwacji w ciągu uczącym. Jest to istotna własność, którą muszą odznaczać się algorytmy operujące na strumieniach danych.

Oprócz poprawności działania algorytmów, w przypadku metod AQ-P1 i AQ-P2, zauważyć można skuteczność stosowania teorii statystycznego uczenia (twierdzenie 2.1), która określa ile należy przyjąć obserwacji, aby w sensie średnim uzyskać niewielki błąd podejmowania decyzji. Dla rozpatrywanego obiektu o trzech wejściach i łącznej liczbie wartości równiej 27, powinno przyjąć się długość okna przesuwnego równą ok. 30. Dla tak dobranej wartości algorytmy uzyskały bardzo dobre wyniki dla każdego kontekstu (wartość kryterium $1 - Q$ pow. 0.9).

Cel badania został osiągnięty. Algorytmy AQ-P1, AQ-P2 i GRI dla każdego kontekstu odtworzyły prawie idealnie własność obiektu i uzyskały bardzo dobre wyniki (wartość zadanego kryterium pow. 0.9).

6.5 Zadanie ekstrakcji wiedzy w przypadku losowym – *Electricity*

Opis zbioru danych

Zbiór danych *Electricity* został po raz pierwszy opisany w [59]. Dotyczy on obserwacji zebranych z *Australian New South Wales Electricity Market*, australijskiego rynku cen elektryczności. Na rynku tym ceny nie są ustalone, ale zmieniają się ze względu na zapotrzebowania i dostępne zasoby. Istnieje kilka głównych czynników wpływających na rynek, np. pogoda, pora dnia, gęstość zaludnienia, rozwój rynku. Innymi słowy, łatwo zauważalne są efekty sezonowości oraz czułości na krótkotrwałe wydarzenia, np. zmiany pogody. Dlatego też podane czynniki, które są często trudne do przewidzenia czy zmierzenia, utoż-

samiać będziemy z **kontekstem**. Natomiast rynek cen elektryczności traktowany może być jako obiekt losowy niestacjonarny ze względu na wpływ ww zmieniających się czynników zewnętrznych.

Oryginalnie zbiór *Electricity* zawiera 45312 obserwacji z okresu maj 1996 – grudzień 1998. Pojedyncza obserwacja składa się z następujących wejść: u^1 – dzień tygodnia, u^2 – okres dnia, u^3 – cena elektryczności w regionie New South Wales, u^4 – zapotrzebowanie na elektryczność w regionie New South Wales, u^5 – cena elektryczności w regionie Victoria, u^6 – zapotrzebowanie na elektryczność w regionie Victoria, oraz u^7 – zaplanowany przepływ elektryczności między regionami.

Wyjście utożsamiane jest ze zmianą ceny związaną ze średnią ruchomą z ostatnich 24 godzin. Wyjście określa odchylenia ceny od średniej z dnia, czyli nie zależy od długoterminowych trendów. Wartości wyjścia są dwie: UP (cena rośnie), DOWN (cena maleje).

Cel badania

Celem badania jest porównanie ze względu na predykcyjny błąd sekwencyjny poprawności działania algorytmów ekstrakcji wiedzy regułowej prezentowanych w niniejszej pracy, tj. AQ-P1, AQ-P2, GRI, z 11 metodami znanymi w literaturze przedmiotu, dla losowego obiektu niestacjonarnego.

Metodyka badania

1. *Przetwarzanie wstępne zbioru danych*
   Część obserwacji, które posiadały brakujące informacje, zostały usunięte ze zbioru danych.

2. *Transformacja danych*
   Ze względu na fakt, że wejścia 3-7 są numeryczne, zastosowano ich dyskretyzację. Ostatecznie zbiory wartości kolejnych wejść były następujące: $K_1 = 7$ – liczba dni, $K_2 = 48$ – pomiar co 1/2 godziny, $K_3 = 10$ – liczba zakresów ceny, $K_4 = 6$ – liczba zakresów zapotrzebowania, $K_5 = 10$ – liczba zakresów ceny, $K_6 = 7$ – liczba zakresów zapotrzebowania, and $K_7 = 7$ – liczba zakresów przepływów elektryczności. Wszystkich wartości – $K = 95$.

3. *Sposób oceny metod*
   Wszystkie metody zostały porównane przy użyciu predykcyjnego błędu sekwencyjnego (1.14)

4. *Ustalenie parametrów*
   Po kilkukrotnym uruchomieniu dokonano wyboru następujących wartości parametrów algorytmów (jedynie w przypadku algorytmów AQ-P1 i AQ-P2 wartości parametrów mogą być niedostateczne ze względu na długi czas działania jednego uruchomienia). Dla AQ-P1 ustalono $C_1 = 1$, $L = 40$. Dla AQ-P2 ustalono $C_2 = 11$, $C_3 = 0.8$, $L = 30$. Natomiast dla klasyfikatora GRI okazało się, że najlepsze rezultaty osiągnięto dla $\beta = 0.1$ i $\gamma = 0.88$.

5. *Przeprowadzenie eksperymentu*
   W eksperymencie założono, że obserwacje pojawiają się w strumieniu danych. Nowo pojawiająca się obserwacja jest najpierw klasyfikowana, a następnie wykorzystywana do uczenia (po klasyfikacji podawana jest prawdziwa wartość wyjścia).

   W eksperymencie algorytmy AQ-P1 i AQ-P2, oraz GRI porównane zostały z następującymi metodami:

   - IB1 (znane też jako 1-NN) z DDM[2] lub EDDM[3] – klasyfikator najbliższego sąsiada z metodą detekcji zmian;
   - J48 z DDM lub EDDM – implementacja drzewa decyzyjnego C4.5 [123] z metodą detekcji zmian;
   - DWM-NB[4] [78] – klasyfikator *ensemble* naiwnych klasyfikatorów bayesowskich, korzystający z idei dynamicznej ważonej większości;
   - PL-NB[5] [7] – szczególny przypadek klasyfikatora *ensemble* naiwnych klasyfikatorów bayesowskich, w którym jeden klasyfikator uczony jest na całym ciągu uczącym, a drugi włącznie na ostatnich obserwacjach;

[2] ***D**rift **D**etection **M**ethod*
[3] ***E**arly **D**rift **D**etection **M**ethod*
[4] ***D**ynamic **W**eighted **M**ajority*
[5] ***P**aired **L**earner*

- SWIM[6] [5] – klasyfikator implementujący boolowską funkcję progową w oparciu o algorytm WINNOW, ale z oknem przesuwnym;
- NB z wykładniczym zapominaniem [149] – estymacja prawdopodobieństw odbywa się w oparciu o macierz powtórzeń, w której zebrane są dane, natomiat zapominanie odbywa się bezpośrednio na macierzy powtórzeń;
- CART [18] z oknem przesuwnym – algorytm drzewa decyzyjnego z oknem przesuwnym;
- Random Forest [17] z oknem przesuwnym – algorytm ensemble drzew decyzyjnych z oknem przesuwnym;
- SVM [157] z oknem przesuwnym oraz następującą funkcją jądra [13]:

$$k(u, v) = 2^{\mathrm{card}\{u \cap v\}} \tag{6.1}$$

gdzie $\mathrm{card}\{u \cap v\}$ – liczba pokrywających się warunków dla reguł u i v.

Uwagi

- Należy zaznaczyć, że w badaniu dodatkowym celem jest sprawdzenie jak proponowane algorytmy radzą sobie w problemie z większą liczbą wejść niż w eksperymencie STAGGER, oraz jak ich jakość wypada na tle innych metod.
- Według wiedzy autora metody SWIM, NB z zapominaniem, CART, Random Forest oraz SVM nie były wcześniej oceniane na podstawie zbioru *Electricity*. Dlatego też wymienione metody zostały zaimplementowane w środowisku Matlab®.
- Wszelkie uzyskane wyniki są przedstawione albo za literaturą (z zaznaczeniem źródła), albo są efektem badań własnych.

Wyniki

Wyniki dla metod AQ-P1, AQ-P2 oraz GRI wraz z wynikami dla metod porównawczych podano w tabeli 6.11. Trzy najlepsze rezultaty pogrubiono. Dodatkowo, dla metody GRI, w tabeli 6.12 przedstawiono wyniki dla różnych wartości parametrów β i γ.

[6] ***S**hifting **WIN**NOW*

Metoda	**Reguły**	**Kryterium** (1.11)	**Źródło**
IB1 with **DDM**	N	0.23	[50]
IB1 with **EDDM**	N	0.14	[8]
J48 with **DDM**	I	0.21	[50]
J48 with **EDDM**	I	0.16	[8]
DWM-NB	N	0.17	[78]
PL-NB	N	0.19	[7]
SWIM	N	**0.11**	[5]
NB with forgetting	N	0.17	[149]
AQ-P1	T	0.19	[145]
AQ-P2	T	0.19	[145]
CART with shifting window	I	0.13	-
Random Forest with shifting window	N	**0.12**	-
SVM with shifting window	N	0.14	-
GRI	T	**0.12**	-

Tabela 6.11: Wyniki dla porównywanych metod i kryterium (1.11) dla przypadku zmian cen na rynku elektryczności. Każda metoda posiada dodatkową informację, czy w podejściu wykorzystywana jest regułowa reprezentacja wiedzy (T - tak, N - nie, I - może być interpretowana jako reguły).

γ	$\beta = 0$	$\beta = 0.1$	$\beta = 0.2$	$\beta = 0.3$	$\beta = 0.4$
1.00	0.25	0.25	0.25	0.28	0.41
0.99	0.20	0.17	0.17	0.19	0.28
0.95	0.18	0.13	0.14	0.16	0.20
0.90	0.18	0.13	0.14	0.15	0.17
0.89	0.21	0.12	0.13	0.14	0.17
0.88	0.21	**0.12**	0.13	0.14	0.16
0.87	0.22	0.12	0.13	0.14	0.16
0.86	0.22	0.13	0.14	0.14	0.17

Tabela 6.12: Wyniki dla algorytmu GRI dla różnych wartości parametrów β i γ oraz dla kryterium (1.11) dla przypadku zmian cen na rynku elektryczności.

Dyskusja

Zbiór danych *Electricity* zawiera dane dotyczące rzeczywistego rynku cen oraz liczba danych jest wystarczająca ze statystycznego punktu widzenia, aby móc porównać metody bez konieczności przeprowadzania testów statystycznych. Analizując wyniki z tabeli 6.11 łatwo zauważyć, że najlepszy wynik osiągnęła metoda SWIM, 0.11, która była nieznacznie, tj. o 0.01, lepsza od klasyfikatora GRI i Random Forest. Bardzo dobry rezultat uzyskała metoda CART, 0.13. Błąd pozostałych algorytmów mieścił się w przedziale od 0.14 do 0.23. Algorytmy AQ-P1 i AQ-P2 uzyskały bardzo zbliżone wyniki (ok. 0.19), które jednak były dość przeciętnymi wartościami.

Zatem można stwierdzić, że klasyfikator GRI osiągnął bardzo dobre wyniki, natomiast algorytmy AQ-P1 i AQ-P2 – przeciętne. Pomimo nienajlepszych rezultatów metod AQ można przypuszczać, że korzystanie z wiedzy regułowej w przypadku rynku cen elektryczności byłoby przydatne dla analityka. Wyekstrahowane reguły mogłyby znacznie ułatwić poznanie prawideł i sytuacji wpływających na zmiany cen.

Warto również bliżej przyjrzeć się wynikom algorytmu GRI. Uzyskał on najlepsze rezultaty dla $\gamma = 0.88$ i $\beta = 0.1$ (patrz tabela 6.12). Parametr γ odpowiada za zapomina-

nie obserwacji. Stosunkowo niska wartość γ daje się łatwo wytłumaczyć silnym wpływem krótkotrwałych trendów, np. pogody, pory roku, awarii. Natomiast parametr β mówi, czy bardziej przydatne są reguły ogólne, czy wyspecjalizowane. Fakt, że β przyjęła bardzo niską wartość oznacza, w przypadku rynku cen elektryczności ważniejsze były reguły mocno wyspecjalizowane. Obserwacja ta również daje się uzasadnić. Ponieważ zmiany cen są zależne od czynników krótkoterminowych, zatem możliwe, że istnieje kilka zdarzeń z kilku dni, które wywołują podobne efekty. Na przykład, w zimie dzień jest krótszy, dlatego w tym okresie rośnie zużycie energii elektrycznej.

Cel badania został osiągnięty. Algorytm GRI osiągnął prawie najlepsze rezultaty w porównaniu ze wszystkimi metodami (tylko nieznacznie gorsze od metody SWIM), natomiast najlepsze dla algorytmów ekstrakcji wiedzy regułowej. Algorytmy AQ-P1 i AQ-P2 uzyskały przeciętne wyniki.

6.6 Zadanie ekstrakcji wiedzy w zastosowaniu do wspomagania przeprowadzenia wywiadu lekarskiego w terapii cukrzycy

Opis problemu

W celu zmniejszenia kosztów leczenia cukrzycy[7] proponowane są tzw. systemy *eZdrowia* [109, 115]. W systemach tego typu dokonywana jest komunikacja między aparaturą pomiarową a serwerem głównym, między pacjentem a lekarzem (poprzez centrum zgłoszeń), między lekarzami i centrum zgłoszeń. Dodatkowo systemy eZdrowia wspomagają pacjenta w zapoznaniu się z chorobą, oraz lekarza w podejmowanie decyzji.

Zagadnienie wspomagania lekarza w przeprowadzaniu wywiadu lekarskiego jest często wskazywane przez praktyków jako kluczowy element w monitorowaniu oraz odpowiedniemu leczeniu cukrzycy [57]. Wywiad lekarski umożliwia poznanie czynników, które wpły-

[7] Cukrzyca jest stanem, który definiuje się jako określony poziom hiperglikemii zwiększający ryzyko uszkodzeń mikronaczyń [57, 67, 165].

wają na rozwój i leczenie choroby, a których nie zawsze można mierzyć wprost. Okazuje się, że w przypadku cukrzycy niezwykle istotny jest sposób odżywiania, aktywność fizyczna, poziom stresu, nawyki żywieniowe, ilość spożywanego alkoholu, itd. [57]. Zazwyczaj pacjenci nie są skłonni do mówienia o intymnych sytuacjach, dlatego też należy wspomóc lekarza poprzez dostarczenie mu wiedzy, dzięki której jego pytania mogą być formułowane bardziej szczegółowo. Przykładowo, w przypadku, gdy okazuje się, że poziom glukozy we krwi zazwyczaj nie jest w normie w soboty rano, przyczyną takiego stanu rzeczy może być ilość spożytego alkoholu dnia poprzedniego. Wówczas zadaniem lekarza jest, aby odpowiednio zinterpretować otrzymaną wiedzę i zadać odpowiednie pytania. Ponadto, pacjenci sami często nie są świadomi, że np. nieregularny tryb życia silnie oddziałuje na ich stan zdrowia i umożliwienie im wglądu do wiedzy wyekstrahowanej na podstawie obserwacji przyczynia się do poprawy i nastawienia do choroby [165].

Wiedza dostarczana lekarzowi czy pacjentowi musi być przedstawiona w postaci, która jest zrozumiała dla człowieka. Przykładem jest zapis logiczny za pomocą reguł decyzyjnych. Ponadto, zapis logiczny można w prosty sposób przeformułować do zdań w języku naturalnym. Dlatego też w niniejszym punkcie przedstawiono zastosowanie metod ekstrakcji wiedzy w postaci logicznej dla systemu wspomagania przeprowadzenia wywiadu lekarskiego w terapii cukrzycy.

Każdy pacjent jest traktowany jako **obiekt niestacjonarny**, na którego oddziałuje **kontekst**. Kontekstem w tym przypadku są czynniki wpływające na stan zdrowia pacjenta, takie jak [57]:

- proces leczenia (np. stosowanie insuliny i innych leków oraz sposób ich przyswajania przez organizm);
- tryb życia (np. aktywność fizyczna, dieta, używki);
- stan psychiczny (np. stres, sen).

Stosowana metoda ekstrakcji wiedzy powinna uwzględniać przyrostowe przetwarzanie danych, jak również mechanizm zapominania tak, aby posiadać aktualną wiedzę o stanie zdrowia pacjenta. Zmiany w przypadku cukrzycy mogą mieć różny charakter, jednak zazwyczaj są to zmiany nagłe, np. z powodu dostosowania diety, rozpoczęcia przyjmowania

leków [57].

W celu sprawdzenia poprawności działania proponowanych w niniejszej pracy algorytmów skorzystano z rzeczywistych danych zebranych przez dra Michaela Kahna, które dotyczą 70 pacjentów i które udostępniane są w repozytorium danych [153]. Diabetycy dokonywali pomiarów automatycznie, przy użyciu odpowiedniego urządzenia, lub ręcznie. Pojedyncza obserwacja składa się z: (i) daty, (ii) czasu pomiaru, (iii) kodu pomiaru, (iv) poziomu glukozy we krwi. Kod pomiaru dotyczył m.in. rodzaju dawki insuliny, czy pomiar był przed posiłkiem czy po, czy po ćwiczeniach, itd. (szczegóły w [153]).

Cel badania

Celem badania jest porównanie ze względu na predykcyjny błąd sekwencyjny poprawności działania algorytmów ekstrakcji wiedzy regułowej prezentowanych w niniejszej pracy, tj. AQ-P1, AQ-P2, GRI, z metodami znanymi w literaturze przedmiotu, tj. CART, Random Forest, SVM.

Metodyka badania

Przyjęto następującą metodykę działania:

1. *Selekcja danych* Cukrzyca jest chorobą, która rozwija się powoli oraz efekty leczenia również są niezauważalne w krótkich okresach czasu. Dlatego też, aby stosowanie podejścia z uczeniem przyrostowym i zapominaniem miało sens, wybrano 10 pacjentów z 70, wśród których najmniejsza liczba obserwacji wynosiła 926 (116 dni leczenia), najdłuższa – 1327 (149 dni leczenia).

2. *Przetwarzanie wstępne i transformacja danych*
   Cechy w oryginalnej postaci nie są do końca przydatne we wspomaganiu przeprowadzenia wywiadu lekarskiego, dlatego też zaproponowano następujące wejścia: u^1 – dzień tygodnia (poniedziałek, wtorek, itd.), u^2 – pora dnia (4:00 do 10:00, 10:00 do 16:00, 16:00 do 22:00, 22:00 do 4:00), u^3 – kod pomiaru (przed posiłkiem, po posiłku, przed dawką insuliny, inne). Liczba wartości cech była następująca: $K_1 = 7$ – liczba dni tygodnia, $K_2 = 4$ – liczba pór dnia, $K_3 = 4$ – liczba kodów. Poza tym, wyjście określa, czy poziom glukozy jest w normie ($y = 1$), czy też nie ($y = 0$). Wartość

wyjścia określana była na podstawie oryginalnego kodu oraz poziomu glukozy we krwi, np. przed posiłkiem dopuszcza się stężenie 80-120 mg/dl, natomiast po posiłku – 80-140 mg/dl

3. *Sposób oceny metod*
   W celu sprawdzenie działania proponowanych w pracy algorytmów wykorzystano predykcyjny błąd sekwencyjny.

4. *Ustalenie parametrów*
   Po kilkukrotnych uruchomieniu algorytmów ustalono następujące wartości parametrów: dla AQ-P1 $L = 16$ (ostatnie dwa dni pomiarów), $C_1 = 1$, dla AQ-P2 $L = 16$, $C_2 = 16$, $C_3 = 0.7$, dla GRI $\gamma = 0.95$ i $\beta = 0.4$.

   Podobnie jak w przypadku zbioru danych *Electricity* przyjęto, że obserwacje przychodzą w strumieniu danych, więc po ich klasyfikacji wykorzystywane są do walidacji oraz uaktualnienia modelu.

5. *Analiza statystyczna wyników*
   Po uzyskaniu wyników dla 10 pacjentów dokonano porównano metod przy użyciu analizy statystycznej (testu t-Studenta).

Uwagi

- Badania, tj. przetwarzanie danych oraz implementacja algorytmów, przeprowadzono w środowisku Matlab®.

Wyniki

Najlepsze uzyskane wyniki dla proponowanych metod, tj. GRI, AQ-P1, AQ-P2, oraz metod porównawczych, tj. CART z oknem przesuwnym, Random Forest z oknem przesuwnym i SVM z oknem przesuwnym, przedstawiono w tabeli 6.13. Dodatkowo dla algorytmu GRI w tabeli 6.14 przedstawiono wyniki dla różnych wartości parametrów β i γ.

	GRI	AQ-P1	AQ-P2	CART	Random Forest	SVM
średnia	**0.10**	0.15	0.16	0.14	0.13	0.13
najgorsza	**0.16**	0.21	0.22	0.20	0.19	0.18
najlepsza	**0.05**	0.10	0.11	0.09	0.09	0.09
odch. stand.	0.04	0.04	0.03	0.04	0.03	0.03

Tabela 6.13: Najlepsze wyniki dla GRI, AQ-P1, AQ-P2 oraz CART, Random Forest i SVM pod względem kryterium (1.14) dla terapii cukrzycy.

γ	$\beta = 0$	$\beta = 0.1$	$\beta = 0.2$	$\beta = 0.3$	$\beta = 0.4$	$\beta = 0.5$	$\beta = 0.6$
1.0	0.13	0.11	0.11	0.11	0.11	0.11	0.12
0.99	0.13	0.12	0.12	0.11	0.11	0.12	0.12
0.98	0.13	0.11	0.11	0.11	0.10	0.11	0.12
0.97	0.14	0.12	0.12	0.11	0.11	0.12	0.12
0.96	0.14	0.12	0.12	0.11	0.11	0.12	0.12
0.95	0.13	0.12	0.11	0.11	**0.10**	0.11	0.12
0.94	0.14	0.13	0.12	0.11	0.11	0.12	0.12
0.93	0.14	0.13	0.12	0.11	0.11	0.12	0.12
0.92	0.14	0.13	0.12	0.11	0.11	0.12	0.12
0.91	0.14	0.13	0.12	0.11	0.11	0.12	0.12
0.90	0.14	0.13	0.12	0.11	0.11	0.12	0.12

Tabela 6.14: Rezultaty uzyskane przez algorytm GRI dla różnych wartości β i γ dotyczącego średniego kryterium (1.14) dla 10 pacjentów we wspomaganiu terapii cukrzycy.

Dyskusja

Analizując otrzymane rezultaty (patrz tabela 6.13) można stwierdzić, że wiedza regułowa jest skuteczna w ocenie stanu pacjenta, tzn. czy poziom glukozy we krwi jest w normie, czy też nie. Algorytmy AQ-P1 i AQ-P2 usyzkały poziom błędu na poziomie ok. 0.15, natomiast algorytm GRI – 0.1. Natomiast zestawiając otrzymane rezultaty z wynikami dla algorytmów porównawczych widać, że najlepszy rezultat uzyskał klasyfikator GRI, tj. 0.10. Algorytmy AQ-P1 i AQ-P2 otrzymały nieznacznie gorsze wyniki od CART (średni błąd 0.14) oraz gorsze od Random Forest i SVM (dla obu średni błąd wyniósł 0.13). Tym niemniej, w celu sprawdzenia, czy różnice są istotne statystycznie, zastosowano test t-Studenta. Test przeprowadzono wyłącznie dla metody GRI, ponieważ dała ona zdecydowanie najlepsze wyniki. Zatem interesuje nas zbadanie hipotezy $H_0 : \mu < \mu_{GRI}$ (test jednostronny), gdzie μ_{GRI} – średnia dla algorytmu GRI, μ – średnia pozostałych metod.

Zatem okazało się, że:

1. W każdym przypadku test na normalność był zdany pomyślnie na poziomie istotności $\alpha = 0.05$. (Uwaga: wykorzystano zaimplementowany test w środowisku Matlab®). W każdym przypadku wartość p była mniejsza od 0.05.

2. Do sprawdzenia równości wariancji wykorzystano statystykę F z 9 stopniami swobody i na poziomie istotności $\alpha = 0.05$, dla której $F(0.05, 9, 9) = 3.18$. Interesuje nas sprawdzenie, czy $\frac{1}{F} < f < F$, gdzie f oznacza stosunek wariancji. Zatem otrzymano:

$$f_{GRI,AQ-P1} = 1.16 \in [0.31, 3.18], \tag{6.2}$$

$$f_{GRI,AQ-P2} = 0.99 \in [0.31, 3.18], \tag{6.3}$$

$$f_{GRI,CART} = 1.14 \in [0.31, 3.18], \tag{6.4}$$

$$f_{GRI,RandomForest} = 1.28 \in [0.31, 3.18], \tag{6.5}$$

$$f_{GRI,SVM} = 1.48 \in [0.31, 3.18]. \tag{6.6}$$

   Ponieważ wszystkie wartości były mniejsze od wartości F, więc można przyjąć, że wariancje są równe w każdym przypadku.

3. Wartość t na poziomie istotności $\alpha = 0.05$ i 9 stopni swobody wynosi $t(0.05, 9) = 2.262$. Hipoteza zerowa może być odrzucona, jeśli wartość $T = \frac{\bar{z}}{S_z/\sqrt{10}}$ jest większa od

t. Otrzymano następujące wyniki:

$$T_{GRI,AQ-P1} = 8.24, \tag{6.7}$$

$$T_{GRI,AQ-P2} = 9.05, \tag{6.8}$$

$$T_{GRI,CART} = 6.02, \tag{6.9}$$

$$T_{GRI,RandomForest} = 7.21, \tag{6.10}$$

$$T_{GRI,SVM} = 5.37. \tag{6.11}$$

We wszystkich przypadkach wartość T była większa od t, więc można odrzucić hipotezę zerową (we wszystkich przypadkach wartość $p \approx 1$).

Zatem ostatecznie można stwierdzić, że metoda GRI dała statystycznie lepsze rezultaty od pozostałych metod.

Cel badania został osiągnięty, ponieważ algorytm GRI uzyskał istotnie najlepsze wyniki w zastosowaniu do predykcji poziomu glukozy w porównaniu z pozostałymi algorytmami. Natomiast algorytmy AQ-P1 i AQ-P2 dały nieznacznie gorsze rezultaty od metod znanych w literaturze przedmiotu.

Na koniec warto przedstawić przykład reguł prezentowanych lekarzowi. Załóżmy, że interesuje nas określenie warunków, dla których poziom glukozy we krwi nie jest w normie. Rozpatrzono pacjenta nr 67, dla którego przedstawiono reguły w *surowej* postaci (patrz 6.16) oraz zaprezentowane w postaci raportu (patrz 6.15). Prezentowane reguły otrzymano za pomocą algorytmu GRI.

$\phi_1 = \Big(\ (u^3 = \text{przed posiłkiem})\ \Big) \Rightarrow (y = \text{nie w normie})$
$\phi_2 = \Big(\ (u^3 = \text{inne})\ \Big) \Rightarrow (y = \text{nie w normie})$
$\phi_3 = \Big(\ (u^3 = \text{po posiłku})\ \Big) \Rightarrow (y = \text{nie w normie})$
$\phi_4 = \Big(\ (u^2 = \text{[22:00-4:00)}\ )\ \Big) \Rightarrow (y = \text{nie w normie})$
$\phi_5 = \Big(\ (u^2 = \text{[22:00-4:00)}\ ) \wedge (u^3 = \text{inne})\ \Big) \Rightarrow (y = \text{nie w normie})$
$\phi_6 = \Big(\ (u^2 = \text{[10:00-16:00)}\ ) \wedge (u^3 = \text{inne})\ \Big) \Rightarrow (y = \text{nie w normie})$
$\phi_7 = \Big(\ (u^2 = \text{[4:00-10:00)}\ ) \wedge (u^3 = \text{inne})\ \Big) \Rightarrow (y = \text{nie w normie})$
$\phi_8 = \Big(\ (u^2 = \text{[16:00-22:00)}\ ) \wedge (u^3 = \text{inne})\ \Big) \Rightarrow (y = \text{nie w normie})$
$\phi_9 = \Big(\ (u^2 = \text{[10:00-16:00)}\ ) \wedge (u^3 = \text{przed posiłkiem})\ \Big) \Rightarrow (y = \text{nie w normie})$
$\phi_{10} = \Big(\ (u^2 = \text{[4:00-10:00)}\ ) \wedge (u^3 = \text{przed posiłkiem})\ \Big) \Rightarrow (y = \text{nie w normie})$
$\phi_{11} = \Big(\ (u^2 = \text{[10:00-16:00)}\ ) \wedge (u^3 = \text{po posiłkiem})\ \Big) \Rightarrow (y = \text{nie w normie})$

Tabela 6.15: Wiedza regułowa dla pacjenta nr 67 dla klasy *poziom glukozy nie w normie*.

Dla pacjenta nr *67* poziom glukozy we krwi jest *nie w normie* szczególnie *przed posiłkiem*, i w *innych* warunkach.

Czasami ma to również miejsce *po posiłku*, rzadziej *między 20:00 a 4:00*.

W niektórych przypadkach dzieje się tak *po posiłku* i *między 10:00 a 16:00*.

Tabela 6.16: Raport na temat stanu zdrowia pacjenta nr 67 w sytuacji, gdy poziom glukozy we krwi nie jest w normie.

Rozdział 7

Uwagi końcowe

W pracy zaproponowano i zbadano algorytmy ekstrakcji wiedzy z uczeniem przyrostowym dla obiektów niestacjonarnych z uwzględnieniem:

- wykrywania zmian kontekstu;
- podejścia z zapominaniem czasowym i wybiórczym;
- podejścia z zapominaniem wykładniczym.

Realizując cel pracy osiągnięto wyniki stanowiące oryginalny wkład autora w dziedzinę uczenia maszynowego. Przeprowadzone badania pozwoliły wskazać na kierunki dalszych prac.

7.1 Oryginalny wkład w dziedzinę ekstrakcji wiedzy dla obiektów niestacjonarnych

Jako elementy nowości w niniejszej pracy należy przede wszystkim wymienić: opracowanie dwóch algorytmów wykrywania zmian kontekstu, opracowanie nowych algorytmów ekstrakcji wiedzy z oknem przesuwnym i zapominaniem wybiórczym, oraz algorytmu ze strojonym modelem. Dalej omówimy krótko każdy z wymienionych elementów.

Algorytmy wykrywania zmiany kontekstu. W pracy przedstawiono podejście z modelowaniem częstościowym do wykrywania zmian kontekstu, w którym wykorzystuje się miarę niepodobieństwa dwóch rozkładów prawdopodobieństw. Zaproponowano nową miarę niepodobieństwa, dla której oszacowano górne ograniczanie prawdopodobieństwa popełnienia błędu. Podobne twierdzenie udowodniono dla miary Lina-Wonga. Drugi z zaproponowanych algorytmów sformułowano w oparciu o modelowanie bayesowskie, w którym zastosowano aproksymację współczynnika Bayesa dla zmiennych dyskretnych z użyciem kryterium Schwarza.

Algorytmy ekstrakcji wiedzy w reprezentacji regułowej. W pracy zaproponowano trzy nowe algorytmy indukcji reguł z zapominaniem z oknem przesuwnym oraz zapominaniem wykładniczym. Zaprezentowano sposób stosowania mechanizmu zapominania wybiórczego poprzez zastosowanie funkcji oceniającej reguły. Przedstawiono nową formę agregacji obserwacji w reprezentacji grafowej. Takie podejście pozwala na stosowanie mechanizmu zapominania ze współczynnikiem zapominania oraz prowadzi do regularyzacji klasy modeli.

Część wyników, które zostały opisane w pracy, opublikowano w następujących pracach: [19], [130], [144], [145], [147], [148], [146].

7.2 Proponowane kierunki dalszych prac

Przeprowadzona analiza przedmiotu oraz otrzymane wyniki prowadzą do wskazania następujących, dalszych kierunków prac:

- Zaproponowanie metody, która umożliwiłaby skrócenie czasu działania (zmniejszenie złożoności obliczeniowej) algorytmu GRI kosztem niewielkiej straty jakości podejmowania decyzji.

- Zastosowanie tzw. uczenia lokalnego [131, 132] dla przyrostowej ekstrakcji wiedzy. Idea polega na tym, że obserwacje są zapominane nie tylko ze względu na moment pojawienia, ale również na *bliskość*, definiowaną wg zadanej metryki, z innymi obserwacjami.

- Zaproponowanie modyfikacji algorytmu GRI w zadaniu grupowania obiektów (ang. *clustering*). Wydaje się, że zastosowanie reprezentacji grafowej jest szczególnie przydatne ze względu na łatwość stosowania mechanizmu zapominania oraz określania zależności między atrybutami.

- Połączenie podejścia z logiczną reprezentacją wiedzy z probabilistycznymi modelami grafowymi w celu przyspieszenia czasu potrzebnego do podejmowania decyzji. Obecnie takie badania są prowadzone, jednak wciąż nie osiągnięto satysfakcjonujących rezultatów [71].

- Połączenie idei algorytmu GRI z modelowaniem bayesowskim w zadaniu predykcji (podejmowania decyzji). Każda reguła może być traktowana jako model, dla którego wyznaczana jest wiarygodność modelu oraz rozkład predykcyjny. Następnie decyzja wyznaczana jest na podstawie mieszaniny rozkładów, tzn. bayesowskiego uśredniania modelu (ang. *Bayesian Model Averaging*) [13].

- Ciąg dalszy badań dla zastosowania w terapii cukrzycy, których efektem byłoby opracowanie modułu wspomagania wywiadu lekarskiego.

Dodatek

Algorytm AQ (ang. *Algorithm Quasi-optimal*) [34, 51, 102] jest algorytmem aproksymacyjnym indukcji reguł, opartym na idei algorytmu zachłannego pokrycia zbioru [37].

Algorytm 7.2.1. Algorithm Quasi-optimal.

Wejście: (i) ciąg uczący $\{(\mathbf{u}_n, y_n)\}_{n=1}^N$, (ii) ciąg dodatkowy $S := \{(\mathbf{u}_n, y_n)\}_{n=1}^N$, (iii) kryterium jakości Q, (iv) $\Phi := \emptyset$.

Wyjście: Zestaw reguł Φ.

Krok 1: Pobierz obserwację ze zbioru S o najniższym indeksie n, $(\mathbf{u}_n, y_n)$, usuń obserwację z S, $S := S \setminus \{(\mathbf{u}_n, y_n)\}$, oraz podziel S na zbiory obserwacji pozytywnych $\bar{S} = \{(\mathbf{u}, y) \in S : y = y_n\}$ i negatywnych $\tilde{S} = \{(\mathbf{u}, y) \in S : y \neq y_n\}$.

Krok 2: Wygeneruj najogólniejsze reguły w oparciu o $(\mathbf{u}_n, y_n)$, które pokrywają obserwacje z $\bar{S}$ i nie pokrywają obserwacji z $\tilde{S}$;

$$\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_J\}.$$

Krok 3: Jeśli $\Lambda = \emptyset$, to $\lambda^* = \left(\bigwedge_{d=1}^D \alpha_d^{u_n^d} \Rightarrow \alpha_{out}^{y_n} \right)$. W przeciwnym razie wybierz regułę o najwyższej wartości kryterium,

$$Q(\lambda^*; \{(\mathbf{u}_n, y_n)\}_{n=1}^N) = \max_{\lambda \in \Lambda} Q(\lambda; \{(\mathbf{u}_n, y_n)\}_{n=1}^N).$$

Krok 4: Dodaj regułę do modelu, $\Phi := \Phi \vee \lambda^*$, usuń ze zbioru S wszystkie te obserwacje, które są pokryte przez λ^*. Jeśli $S \neq \emptyset$, to idź do kroku 1.

Krok 5: Zwróć model Φ i STOP.

Bibliografia

[1] Adams R. P., MacKay D. J., *Bayesian online changepoint detection*, Raport techniczny, University of Cambridge, Cambridge, UK, 2007, arXiv:0710.3742v1 [stat.ML]. [cytowanie na str. 19, 70, 74]

[2] Andersen T. L., Martinez T. R., *NP-completeness of minimum rule sets*, Proceedings of the 10th International Symposium on Computer and Information Sciences, 411–418, 1995. [cytowanie na str. 26, 27, 46]

[3] Angluin D., *Computational learning theory: Survey and selected bibliography*, Proceedings of the Twenty-Fourth Annual ACM Symposium on Theory of Computing, New York, NY, USA, 351–369, 1992. [cytowanie na str. 18]

[4] Arena Simulation Software, `http://http://www.arenasimulation.com/`. [cytowanie na str. 79]

[5] Auer P., Warmuth M. K., *Tracking the best disjunction*, Machine Learning, 32:127–150, 1998. [cytowanie na str. 95, 96]

[6] Bach S. H., Maloof M. A., *A Bayesian approach to concept drift*, (red.) J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. Zemel, A. Culotta, w: Advances in Neural Information Processing Systems 23, 127–135, 2010. [cytowanie na str. 31]

[7] Bach S. H., Maloof M. A., *Paired learners for concept drift*, Proceedings of Eighth IEEE International Conference on Data Mining, 23–32, 2008. [cytowanie na str. 94, 96]

[8] Baena-García M., del Campo-Ávila J., Fidalgo R., Bifet A., Gavalda R., Morales-Bueno R., *Early drift detection method*, Proceedings of ECML PKDD 2006 Workshop on Knowledge Discovery from Data Streams, Berlin, Germany, 2006. [cytowanie na str. 19, 96]

[9] Bartlett P. L., Helmbold D. P., *Tracking a drifting concept in a changing environment*, Raport techniczny UCSC-CRL-98-12, University of California, Santa Cruz, California, March 1998. [cytowanie na str. 19]

[10] Basseville M., Nikiforov I. V., *Detection of Abrupt Changes - Theory and Application*, Prentice-Hall, Englewood Cliffs, N.J., 1993. [cytowanie na str. 15, 19]

[11] Becker J. M., *Topics in incremental learning of disciminant descriptions*, Raport techniczny, University of Illinois, Illinois, USA, 1985. [cytowanie na str. 7, 26]

[12] Berger J. O., *Statistical Decision Theory and Bayesian Analysis*, Sprigner-Verlag, New York, Berlin, Heidelberg, Tokyo, 1985. [cytowanie na str. 30, 33, 39]

[13] Bishop C. M., *Pattern Recognition and Machine Learning*, Springer-Verlag, Singapore, 2006. [cytowanie na str. 2, 4, 14, 26, 30, 33, 39, 67, 95, 108]

[14] Black M., Hickey R., *Learning classification rules for telecom customer call data under concept drift*, Soft Computing, 8:102–108, 2002. [cytowanie na str. 1, 26]

[15] Blumer A., Ehrenfeucht A., Haussler D., Warmuth M. K., *Learnability and the vapnik-chervonenkis dimension*, Journal of the Association for Computing Machinery, 36(4):929–965, 1989. [cytowanie na str. 24, 26]

[16] Bocheński J.M., *Współczesne metody myślenia*. W Drodze - Wydawnictwo Polskiej Prowincji Dominikanów, Poznań, 1993. ("Modern Methods of Reasoning"). [cytowanie na str. 6]

[17] Breiman L., *Random forests*, Machine Learning, 45(1):5–32, 2001. [cytowanie na str. 95]

[18] Breiman L., Friedman J., Olshen R., Stone P., *Classification and Regression Trees*, Wadsworth, Belmont, 1984. [cytowanie na str. 95]

[19] Brzostowski K., Tomczak J. M., *Wspomaganie przeprowadzenia wywiadu lekarskiego dla systemu zarządzającego terapią cukrzycy*, XVII Krajowa Konferencja Automatyki, Kielce-Cedzyna, 2011, ("Automatic assistance in conducting medical interview in expert system for diatebes therapy"). [cytowanie na str. 1, 26, 69, 107]

[20] Brzostowski K., Tomczak J. M., Rekuć W., Sobecki J., *Service discovery approach based on rough sets for soa systems*, (red.) N. T. Nguyen, A. Zgrzywa, A. Czyżewski, w: Advances

in multimedia and network information system technologies, Springer, Berlin, Heidelberg, 131–141, 2010. [cytowanie na str. 1]

[21] Bubnicki Z., *Identification of Control Plants*, Elsevier, Oxford, Amsterdam, New York, 1980. [cytowanie na str. 2, 3, 5, 9, 15, 17, 65]

[22] Bubnicki Z., *Wstęp do systemów ekspertowych*, PWN, Warszawa, 1990, ("Introduction to the Expert Systems"). [cytowanie na str. 2, 4, 23, 25, 26, 28]

[23] Bubnicki Z., *Knowledge-based approach as a generalization of pattern recognition problems methods*, Systems Science, 19(2):5–20, 1993. [cytowanie na str. 4, 23, 25, 28]

[24] Bubnicki Z., *Podstawy informatycznych systemów zarządzania*, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 1993, ("Principles of the Information and Management Systems"). [cytowanie na str. 1, 2]

[25] Bubnicki Z., *Knowledge updating in a class of knowledge-based learning control systems*, Systems Science, 23(4):19–36, 1997. [cytowanie na str. 4, 47, 50]

[26] Bubnicki Z., *Learning algorithms for a class of knowledge-based systems with dynamical knowledge representation*, Systems Science, 26(1):15–27, 2000. [cytowanie na str. 47]

[27] Bubnicki Z., *Analysis Decision Making in Uncertain Systems*, Springer-Verlag, London, 2004. [cytowanie na str. 2, 4, 5, 10, 25, 28]

[28] Bubnicki Z., *Modern Control Theory*, Springer-Verlag, Berlin, 2005. [cytowanie na str. 1]

[29] Carbonell T. J., Michalski R. S., Mitchell T. M., *An Overview of machine learning*, (red.) R. Michalski, T. J. Carbonell, T. M. Mitchell, w: Machine Learning: An Artificial Intelligence Approach, Palo Alto, TIOGA Publishing Co., 3–23, 1983. [cytowanie na str. 3, 4, 5]

[30] Cervone G., Michalski R., *Modeling user behavior by integrating aq learning with a database: Initial results*, Proceedings of the IIS-02 Eleventh International Symposium on Intelligent Information Systems, Sopot, Polska, 2002. [cytowanie na str. 1, 26]

[31] Cherkassky V., Mulier F., *Learning from Data. Concepts, Theory, and Methods*, John Wiley & Sons, IEEE Press, 2007. [cytowanie na str. 26]

[32] Chib S., *Estimation and comparison of multiple change-point models*, Journal of Econometrics, 86:221–241, 1998. [cytowanie na str. 70, 74]

[33] Chmielewski A., *Filozofia Poppera. Analiza krytyczna*, Aletheia, 2002, ("The Popper's Philosophy. A Critical Analysis"). [cytowanie na str. 7]

[34] Cichosz P., *Systemy uczące się*, PWN, Warszawa, 1998, ("Learning Systems"). [cytowanie na str. 2, 4, 5, 6, 7, 18, 25, 26, 49, 109]

[35] Cios K. J., Pedrycz W., Świniarski R. A., Kurgan Ł.A., *Data Mining. A Knowledge Discovery Approach*, Springer-Verlag, New York, 2005. [cytowanie na str. 4, 5, 18]

[36] Clark P., Niblett T., *The CN2 induction algorithm*, Machine Learning, 3:261–283, 1989. [cytowanie na str. 18]

[37] Cormen T. H., Leiserson C. E., Rivest R. L., *Wprowadzenie do algorytmów*, WNT, Warszawa, 2001, ("Introduction to Algorithms"). [cytowanie na str. 109]

[38] Cypkin J. Z., *Adaptacija i obučenie v avtomatičeskich sistemach*, Nauka, Moskwa, 1968, ("Adaptation and Learning in Automation Systems"). [cytowanie na str. 8, 16]

[39] Devroye L., Győrfi L., Lugosi G., *A Probabilistic Theory of Pattern Recognition*, Springer-Verlag, New York, 1991. [cytowanie na str. 27, 33, 34]

[40] Diestel R., *Graph Theory*, Springer-Verlag, New York, 2000. [cytowanie na str. 54, 55]

[41] Dragomir S. S., Šunde J., Buşe C., *Divergence measures based on the shannon entropy*, Tamsui Oxford Journal of Mathematical Sciences, 37(1):145–151, 1991. [cytowanie na str. 35, 36, 37]

[42] Dries A., Rűckert U., *Adaptive concept drift detection*, Statistical Analysis for Data Mining, 2(5-6):311–327, 2009. [cytowanie na str. 19]

[43] European Commission. *From grids to service-oriented knowledge utilities. a critical infrastructure for business the citizen in the knowledge society*, Raport techniczny, 2006. [cytowanie na str. 1, 77]

[44] Fayyad U., Piatetsky-Shapiro G., Smyth P., *From data mining to knowledge discovery in databases*, Artificial Intelligence Magazine, 17(3):37–54, 1996. [cytowanie na str. 1, 5, 26]

[45] Fearnhead P., *Exact and efficient bayesian inference for multiple changepoint problems*, Statistics and Computing, 16(2):203–213, 2006. [cytowanie na str. 19, 70, 74]

[46] Ferrer-Troyano F., Aguilar-Ruiz J. S., Riquel J. C., *Incremental rule learning based on example nearness from numerical data streams*, Proceedings of the 2005 ACM symposium on Applied computing SAC '05, 568–572, New York, NY, USA, 2005. [cytowanie na str. 19]

[47] Frawley W. J., Piatetsky-Shapiro G., Matheus C. J., *Knowledge discovery in databases: An Overview*, Artificial Intelligence Magazine, 13(3):57–70, 1992. [cytowanie na str. 1, 2]

[48] Gama J., Sebastião S., Rodrigues P., *Issues in evaluation of stream learning algorithms*, Proceedings of the 15th ACM SIGKDD Int. Conf. on KDD, 329–338, 2009. [cytowanie na str. 16]

[49] Gama J., Klinkenberg R., Aguilar J., *Knowledge discovery from data streams*, (red.) J. Gama, R. Klinkenberg, J. Aguilar, w: The Fourth International Workshop on Knowledge Discovery from Data Streams, 2006. [cytowanie na str. 1]

[50] Gama J., Medas P., Castillo G., Rodrigues P., *Learning with drift detection*, Lecture Notes in Computer Science, 3171:66–112, 2004. [cytowanie na str. 19, 96]

[51] Gatnar E., *Symboliczne metody klasyfikacji danych*, PWN, Warszawa, 1998, ("Symbolic Methods For Data Classification"). [cytowanie na str. 2, 3, 4, 26, 49, 109]

[52] Georgii E., Tsuda K., Schőlkopf B., *Multi-way set enumeration in weight tensors*, Machine Learning, 82:123–155, 2011. [cytowanie na str. 18]

[53] Ghosh J. K., Delampady M., Samanta T., *An Introduction to Bayesian Analysis: Theory and Methods*, Sprigner-Verlag, New York, 2006. [cytowanie na str. 39, 40, 42]

[54] Gniedenko B. W., Kowalenko I. N., *Wstęp do teorii obsługi masowej*, PWN, Warszawa, 1975, ("Introduction to the Queueing Theory") [cytowanie na str. 80]

[55] Green P. J., *Reversible jump markov chain monte carlo computation and bayesian model determination*, Biometrika, 82(4):711–732, 1995. [cytowanie na str. 70, 74]

[56] Grzech A., *Sterowanie ruchem w sieciach teleinformatycznych*, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 2002, ("Teletraffic Control in Teleinformatics Networks"). [cytowanie na str. 1]

[57] Grzeszczak W. et al., *Zalecenia kliniczne dotyczące postępowania u chorych na cukrzycę 2010. Stanowisko polskiego towarzystwa diabetologicznego*, Pismo Polskiego Towarzystwa Diabetologicznego, Tom 11, Supl. A, 2010, ("Clinical recommendations for diabetics 2010. A Standpoint of Polish Diabetes Association") [cytowanie na str. 98, 99, 100]

[58] Gustafsson F., *Adaptive Filtering and Change Detection*, John Wiley & Sons, 2000. [cytowanie na str. 15, 19, 31]

[59] Harries M. B., *SPLICE-2 comparative evaluation: Electricity pricing*, Raport techniczny UNSW-CSETR-9905, 1999. [cytowanie na str. 69, 92]

[60] Harries M. B., Sammut C., Horn K., *Extracting hidden context*, Machine Learning, 32:101–126, 1998. [cytowanie na str. 8, 19]

[61] Haussler D., *Quantifying inductive bias: AI learning algorithms and valiant's learning framework*, Artificial Intelligence Magazine, 36:177–221, 1988. [cytowanie na str. 18]

[62] Haussler D., Littlestone N., Warmuth M. K., *Predicting $\{0,1\}$-functions on randomly drawn points*, Information and Computation, 115:248–292, 1994. [cytowanie na str. 18, 26]

[63] Herrera F., Carmona C. J., Gonzalez P., del Jesus M. J., *An overview on subgroup discovery: foundations and applications*, Knowledge and Information Systems, 2010. [cytowanie na str. 58]

[64] Holder L. B., Cook D. J., *Graph-based data mining*, (red.) J. Wang, w: Encyclopedia of Data Warehousing and Mining, Idea Group Publishing, 2005. [cytowanie na str. 18]

[65] Hulten G., Spencer L., Domingos P., *Mining time changing data streams*, Proceedings of the seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Dining, San Francisco, California, 97–106, 2001. [cytowanie na str. 19]

[66] Inokuchi A., Washio T., Motoda H., *Complete mining of frequent patterns from graphs: Mining graph data*, Machine Learning, 50:321–354, 2003. [cytowanie na str. 18]

[67] International Diabetes Federation and Federation of European Nurses in Diabetes, *The policy puzzle: Towards benchmarking in the eu 25*, Raport IDF/FEND, 2005, `http://www.idf.org/webdata/docs/idf-europe/DiabetesReport2005.pdf`. [cytowanie na str. 98]

[68] Jagielski J., *Inżynieria wiedzy w systemach ekspertowych*, Oficyna Wydawnicza Uniwersytetu Zielonogórskiego, Zielona Góra, 2001, ("Knowledge Engineering in the Expert Systems"). [cytowanie na str. 2, 4, 26, 28]

[69] Jarret R. G., *A note on the intervals between coal-mining disasters*, Biometrika, 66:71–83, 1979. [cytowanie na str. 69]

[70] Jordan M. I., *Graphical models*, Statistical Science, 19(1):140–155, 2004. [cytowanie na str. 4]

[71] Jordan M. I., *Bayesian nonparametric learning: Expressive priors for intelligent systems*, (red.) R. Dechter, H. Geffner, J. Halpern, w: Heuristics, Probability and Causality: A Tribute to Judea Pearl, College Publications, 159–178, 2010. [cytowanie na str. 5, 108]

[72] Józefczyk J., *Systemy z reprezentacją wiedzy. Systemy ekspertowe*, w: Problemy automatyki i informatyki, Zakład Narodowy im. Ossolińskich, Wrocław, 71–81, 1998, ("Knowledge-based systems. Expert systems"). [cytowanie na str. 23, 26]

[73] Kailath T., *The divergence and Bhattacharyya distance measures in signal selection*, IEEE Transactions on Communication Techniques, 15(1):52–60, 1967. [cytowanie na str. 35, 36]

[74] Kass R. E., Raftery A. E., *Bayes factors*, Journal of the American Statistical Association, 90(430):773–795, 1995. [cytowanie na str. 40, 42, 130]

[75] Kaufmann K. A., Michalski R. S., *From data mining to knowledge mining*, (red.) C. Rao, J. Solka, E. Wegman, w: Handbook in Statistics, volume 24 of *Data Mining and Data Visualization*, Elsevier, North Holland, 47–75, 2005. [cytowanie na str. 7]

[76] Kazakos D., Cotsidas T., *A Decision theory approach to the approximation of discrete probability densities*, IEEE Transactions on Paternn Analysis and Machine Intelligence, PAMI-2(1):61–67, 1980. [cytowanie na str. 33, 34, 35, 36]

[77] Kifer D., Ben-David S., Gehrke J., *Detecting change in data streams*, Proceedings of the 30th VLDB Conference, Toronto, Canada, 180–191, 2004. [cytowanie na str. 19]

[78] Kolter J. Z., Maloof M. A., *Using additive expert ensembles to cope with concept drift*, Proceedings of the Twenty-second International Conference on Machine Learning, New York, NY, 449–456, 2005. [cytowanie na str. 94, 96]

[79] Kotsiantis S. B., *Supervised machine learning: A Rewiev of classification techniques*, Informatica, 31(3):249–268, 2007. [cytowanie na str. 2, 24, 25, 26]

[80] Kotsiantis S. B., Kanellopoulos D., Pintelas P. E., *Data preprocessing for supervised leaning*, International Journal of Computer Science, 1:111–117, 2006. [cytowanie na str. 6]

[81] Koychev I., *Learning about user in the presence of hidden context*, Proceedings of the UM2001 Workshop on Machine Learning for User Modeling, 49–58, 2001. [cytowanie na str. 8, 26]

[82] Koychev I., Schwab I., *Adaptation to drifting user's interests*, ECML2000 Workshop: Machine Learning in New Information Age, 39–46, 2000. [cytowanie na str. 1, 26]

[83] Krzyśko M., Wołyński W., Górecki T., Skorzybut M., *Systemy uczące się. Rozpoznawanie wzorców, analiza skupień i redukcja wymiarowości*, WNT, Warszawa, 2008, ("Learning Systems. Pattern Recognition, Data Clustering, and Dimensionality Reduction"). [cytowanie na str. 4, 26]

[84] Kubat M., *Introduction to machine learning*, Proceedings of Advanced Topics in Artificial Intelligence, 104–138, 1992. [cytowanie na str. 19]

[85] Kubat M., *A machine learning-based approach to load balancing in computer networks*, Cybernetics and Systems: An International Journal, 23(3):389–400, 1992. [cytowanie na str. 1, 26]

[86] Kurzyński M., *Rozpoznawanie obrazów. Metody statystyczne*, Oficyna Wydawnicza PWr, 1997, ("Pattern Recognition. The Statistical Approach"). [cytowanie na str. 13, 28]

[87] Langley P., Simon H. A., *Applications of machine learning rule induction*, *Communications of the ACM*, 38(11):54–64, 1995. [cytowanie na str. 1, 6, 26]

[88] Last M., *Online classification of nonstationary data streams*, Intelligent Data Analysis, 6:129–147, 2002. [cytowanie na str. 20]

[89] Last M., Klein Y., Kandel A., *Knowledge discovery in time series databases*, IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, 31:160–169, 2001. [cytowanie na str. 20]

[90] Lavrač N., *Machine learning for data mining in medicine*, Lecture Notes in Artificial Intelligence, 1620:47–62, 1999. [cytowanie na str. 1, 26]

[91] Lavrač N., Džeroski S., *Inductive Logic Programming: Techniques and Applications*, Ellis Horwood, New York, 1994. [cytowanie na str. 1, 4, 7, 18]

[92] Lee C.-B., *Bayesian analysis of a change-point in exponential families with applications*, Computational Statistics & Data Analysis, 27:195–208, 1998. [cytowanie na str. 70, 74]

[93] Ligęza A., *Logical Foundations for Rule-Based Systems*, Uczelniane Wydawnictwa Naukowo-Dydaktyczne Akademii Górniczo-Hutniczej w Krakowie, Kraków, 2005. [cytowanie na str. 4, 23]

[94] Lin J., *Divergence measures based on the shannon entropy*, IEEE Transactions on Information Theory, 37(1):145–151, 1991. [cytowanie na str. 36, 37]

[95] Lite Technologies. *Web server performance comparison: Litespeed 2.0 vs.*, `http://litespeedtech.com/web-server-performance-comparison-litespeed-2.0-vs.html`. [cytowanie na str. 81]

[96] Mańczak K., Nahorski Z., *Komputerowa identyfikacja obiektów dynamicznych*, PWN, Warszawa, 1983, ("Computer-based Identification for Dynamic Objects"). [cytowanie na str. 7, 16]

[97] Maloof M. A., *Progressive partial memory learning*, Rozprawa doktorska, School of Information Technology and Engineering, George Mason University, Fairfax, VA, 1996. [cytowanie na str. 26]

[98] Maloof M. A., Michalski R. S., *Selecting examples for partial memory learning*, Machine Learning, 41(1):27–52, 1999. [cytowanie na str. 7, 8, 17, 90]

[99] Maloof M. A., Michalski R. S., *Incremental learning with partial instance memory*, Lecture Notes on Artificial Intelligence, 2366:16–27, 2002. [cytowanie na str. 26]

[100] Matwin S., Kubat M., *The role of context in concept learning*, Proceedings of the ICML-96 Workshop on Learning in Context-Sensitive Domains, 1–5, 1996. [cytowanie na str. 8]

[101] van der Mei R. D., Hariharan R., Reeser P. K. Web server performance modelling. *Telecommunication Systems*, 16(3-4):316–378, 2001. [cytowanie na str. 77, 81]

[102] Michalski R. S., *On the quasi-minimal solution of the general covering problem*, Proceedings of the V International Symposium on Information Processing, Yugoslavia, 125–128, 1969. [cytowanie na str. 4, 18, 23, 109]

[103] Michalski R. S., *A theory and methodology of inductive learning*, (red.) R. S Michalski, T. J. Carbonell, T. M. Mitchell, Machine Learning: An Artificial Intelligence Approach, TIOGA Publishing Co, Palo Alto, 83–134, 1983. [cytowanie na str. 6, 18, 25]

[104] Michalski R. S., *Learning strategies and automated knowledge acquisition: An Overview*, Report 926, Department of Computer Science, Univeristy of Illinois, Urbama, Illinois, November 1984. [cytowanie na str. 1]

[105] Michalski R. S., Larson J. B., *Incremental generation of VL_1 hypothesis: The underlying methodology. The description of program $AQ11$*, Reports of the intelligent systems group, University of Illinois, 1983. [cytowanie na str. 18]

[106] Mitchell T., *Machine Learning*. McGraw Hill, 1997. [cytowanie na str. 2, 4, 5, 18, 59]

[107] Mitra S., Pal S., Mitra P., *Data mining in soft computing framework: A Survey* IEEE Transactions on Neural Networks, 13(1):3–14, 2002. [cytowanie na str. 5]

[108] Moreno E., Casella G., Garcia-Ferrer A., *An Objective Bayesian analysis of the change point problem*, Stochastic Environmental Research and Risk Assessment, 19(3):191–204, 2005. [cytowanie na str. 70, 74]

[109] Mougiakakou S. G. et al., *Smartdiab: A Communication and information technology approach for the intelligent monitoring, management and follow-up of type 1 diabetes patients*, IEEE Transactions on Information Technology in Biomedicine, 14:622–633, 2010. [cytowanie na str. 98]

[110] Natschlåger T., Schmitt M., *Exact VC-dimension of boolean monomials*, Information Processing Letters, 59:19–20, 1996. [cytowanie na str. 24, 27]

[111] van Ness J. W., *Dimensionality and the classification performance with independent coordinates*, IEEE Transactions on Systems, Man, and Cybernetics, SMC-7:560–564, 1977. [cytowanie na str. 33, 36]

[112] Nguyen N. T., *Advanced Methods for Inconsistent Knowledge Management*. Springer-Verlag, London, 2008. [cytowanie na str. 26]

[113] Niederliński A., Mościński J., Ogonowski Z., *Regulacja adaptacyjna*, PWN, Warszawa, 1995, ("Adaptive Control Theory"). [cytowanie na str. 7, 8, 65]

[114] Pal N. R., Jain L., *Preface*, (red.) L. Jain, N.R. Pal, w: Advanced Techniques in Knowledge Discovery Data Mining, Springer-Verlag, London, 2005. [cytowanie na str. 1]

[115] Pantelopoulos A., Bourbakis N. G., *A Survey on wearable sensor-based systems for health monitoring and prognosis*, IEEE Transactions on Antennas & Propagation Magazine, 44(2):143–153, 2010. [cytowanie na str. 98]

[116] Pawlak Z., *Systemy informacyjne. Podstawy teoretyczne*, WNT, Warszawa, 1983, ("Information Systems. Theoretical Principles"). [cytowanie na str. 4]

[117] Pawlak Z., *In pursuit of patterns in data reasoning from data - the rough set way*, Lecture Notes in Computer Science, 2475:1–9, 2002. [cytowanie na str. 18, 49, 58, 62]

[118] Pawlak Z., *Data analysis and flow graphs*, Journal of Telecommunications and Information Technology, 3:1–5, 2004. [cytowanie na str. 18, 62, 63]

[119] Paxson V., Floyd S., *Wide-area traffic: The failure of poisson modeling*, IEEE/ACM Transactions on Networking, 3(3):226–244, 1995. [cytowanie na str. 80]

[120] Popper K., *Logika odkrycia naukowego*, Aletheia, 2002, ("Logic of Scientific Discovery"). [cytowanie na str. 6, 7]

[121] Potts J., Cook D. J., Holder L. B., *Learning from supervised graphs*, Studies in Computational Intelligence, 52:183–201, 2007. [cytowanie na str. 18]

[122] Prusiewicz A., Zięba M., *The proposal of service oriented data mining system for solving real-life classification and regression problems*, (red.) L. Camarinha-Matos, w: Technological Innovation for Sustainability, volume 349 of *IFIP Advances in Information and Communication Technology*, 83–90, Springer-Verlag, Boston, 2011. [cytowanie na str. 81]

[123] Quinlan J. R., *Learning efficient classification procedures and their application to chess end games*, (red.) R. S. Michalski, J. G. Carbonell, T. M. Mitchell, w: Machine learning: An Artificial intelligence approach, CA: Morgan Kaufmann, San Mateo, 463–482, 1983. [cytowanie na str. 18, 94]

[124] Raftery A. E., Akman V. E., *Bayesian analysis of a poisson process with a change-point*, Biometrika, 73(1):85–89, 1986. [cytowanie na str. 69, 70, 74]

[125] Rasiowa H., *Wstęp do matematyki współczesnej*, PWN, Warszawa, 1971, ("An Introduction to the Modern Mathmatics"). [cytowanie na str. 23, 24, 25]

[126] Rasmussen C. E., Williams C. K. I., *Gaussian Processes for Machine Learning*, MIT Press, Cambridge, Massachusetts, USA, 2006. [cytowanie na str. 5, 19]

[127] Reinke R. E., Michalski R. S., *Incremental learning of concept description: A Method and experimental results*, (red.) J.E.Hayes, D. Michie, J. Richards, w: Machine Intelligence, Clarendon Press, Oxford, 263–288, 1988. [cytowanie na str. 18]

[128] Rűckert U., Kramer S., *A Statistical approach to rule learning*, Proceedings of the 23rd International Conference on Machine Learning, New York, 785–792, 2006. [cytowanie na str. 18]

[129] Rutkowski L., *Metody i techniki sztucznej inteligencji*, PWN, Warszawa, 2006, ("Artificial Intelligence Methods and Techniques"). [cytowanie na str. 3, 4]

[130] Rygielski P., Tomczak J. M., *Context change detection for resource allocation in service-oriented systems*, Lecture Notes in Computer Science, Lecture Notes in Artificial Intelligence, 6882:591-600, 2011. [cytowanie na str. 1, 78, 107]

[131] Salganicoff M., *Density-adaptive learning and forgetting*, Proceedings of the Tenth International Conference on Machine Learning, Vol. A3, CA: Morgan Kaufmann, San Francisco, USA, 276–283, 1993. [cytowanie na str. 8, 107]

[132] Salganicoff M., *Tolerating concept and sampling shift in lazy learning using prediction error context switching*, Artificial Intelligence Review, 11:133–155, 1997. [cytowanie na str. 7, 8, 107]

[133] Sas A., Świątek P., Tomczak J. M., Zięba M., *Multistage parallel processing of connections in the network node: Simulation model*, (red.) L. Borzemski, w: Information Systems Architecture Technology: Designing, Development Implementation of Information Systems, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 135–143, 2008. [cytowanie na str. 1]

[134] Schlimmer J., Granger Jr. G., *Incremental learning from noisy data*, Machine Learning, 8:317–354, 1986. [cytowanie na str. 20, 69, 89]

[135] Schwarz G., *Estimating the dimension of a model* The Annals of Statistics, 6(2):461–464, 1978. [cytowanie na str. 43]

[136] Sebastiao R., Gama J., *Change detection in learning histograms from data streams*, Lecture Notes in Artificial Intelligence, 4874:112–123, 2007. [cytowanie na str. 19]

[137] Shi Z., *Principles of Machine Learning*, International Academic Publishers, Beijing, 1992. [cytowanie na str. 6]

[138] soapUI, `http://www.soapui.org/`. [cytowanie na str. 82]

[139] Sobczak W., Malina W., *Metody selekcji i redukcji informacji*, WNT, Warszawa, 1985, ("Methods for Information Selection and Reduction"). [cytowanie na str. 3, 6, 35, 36]

[140] Sobecki J., Tomczak J. M., *Student courses recommendation using ant colony optimization*, Lecture Notes in Computer Science, Lecture Notes in Artificial Intelligence, 5991:124–133, 2010. [cytowanie na str. 1]

[141] Stierli I., *Entropy-based, semi-dynamic regulation of incremental algorithms in the case of instantaneous concept drifts*, Praca magisterska, University of Zurich, Zurich, Switzerland, 2005. [cytowanie na str. 19, 31, 32]

[142] Świątek J., *Some problems of identification for relation systems*, Systems Science, 15(1):26–34, 1989. [cytowanie na str. 10]

[143] Świątek J., *Wybrane zagadnienia identyfikacji statycznych systemów złożonych*, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 2009, ("Selected Issues on Identification of Static Complex Systems"). [cytowanie na str. 2]

[144] Świątek J., Brzostowski K., Tomczak J. M., *Computer aided physician interview for remote control system of diabetes therapy*, InterSymp 2011, 23rd International Conference on System Research, Informatics & Cybernetics, Baden-Baden, 2011. [cytowanie na str. 69, 107]

[145] Tomczak J. M., *Algorithms for Knowledge Extraction Using Relation Identification. A New Approach*, Lambert Academic Publishing, Saarbrűcken, 2010. [cytowanie na str. 1, 10, 96, 107]

[146] Tomczak J. M., *On-line Change Detection for Resource Allocation in Service-Oriented Systems*, Volume 372 of *IFIP Advances in Information and Communication Technology*, Springer Verlag, 2012. (W druku). [cytowanie na str. 1, 107]

[147] Tomczak J. M., Brzostowski K., Świątek J., *Knowledge extraction using shifting window from non-stationary datastreams*, (red.) A. Grzech, w: Information systems architecture and technology: Networks and networks' services, Oficyna Wydawnicza PWr., Wrocław, 321–331, 2010. [cytowanie na str. 107]

[148] Tomczak J. M., Gonczarek A., *Decision rules extraction from data stream in the presence of changing context for diabetes treatment*, Knowledge and Information Systems (zaakceptowane) [cytowanie na str. 1, 26, 69, 107]

[149] Tomczak J. M., Świątek J., *Bayesian classifiers with incremental learning for nonstationary datastreams*, (red.) J. Drapała, A. Grzech, P. Świątek, w: Advances in Systems Science, EXIT, Warszawa, 251–260, 2010. [cytowanie na str. 95, 96]

[150] Tomczak J. M., Świątek J., *Personalisation in service-oriented systems using markov chain model and bayesian inference*, (red.) L. M. Camarinha-Matos, w: Technological innovation for sustainability, Vol. 349 of *IFIP Advances in Information and Communication Technology*, Springer-Verlag, Heidelberg, 91–98, 2011. [cytowanie na str. 1, 16]

[151] Torgo L., *Controlled redundancy in incremental rule learning*, Lecture Notes in Artificial Intelligence, 667:185–195, 1993. [cytowanie na str. 18, 28]

[152] Tsymbal A., *The problem of concept drift: Definitions and related work*, Raport techniczny, Trinity College Dublin, 2004. [cytowanie na str. 2, 8]

[153] UCI Machine Learning Repository, `http://archive.ics.uci.edu/ml/datasets.html`. [cytowanie na str. 69, 100]

[154] Utgoff P. E., *Incremental induction of decision trees*, Machine Learning, 4:161–186, 1989. [cytowanie na str. 18]

[155] Valiant L. G., *A Theory of the learnable*, Communications of ACM, 27:1134–1142, 1984. [cytowanie na str. 18]

[156] Vapnik V. N., *The Statistical Learning Theory*, John Wiley & Sons, New York/ Chichester/ Weinheim/ Brisbane/ Singapore/ Toronto, 1998. [cytowanie na str. 5, 7, 15, 26]

[157] Vapnik V. N., *An Overview of statistical learning theory*, IEEE Transactions on Neural Networks, 10(5):988–999, 1999. [cytowanie na str. 26, 95]

[158] Vorburger P., Bernstein A., *Entropy-based concept shift detection*, Proceedings of the Sixth International Conference on Data Mining, 1113–1118, 2006. [cytowanie na str. 19]

[159] Washio T., Motoda H., *State of the art of graph-based data mining*, ACM SIGKDD Explorations Newsletter, 5:59–68, 2003. [cytowanie na str. 18]

[160] Webb G, Pazzani M., Billsus D., *Machine learning for user modeling*, User Modeling User-Adapted Interaction, 11:19–29, 2001. [cytowanie na str. 1, 26]

[161] Widmer G., *Tracking context changes through meta-learning*, Machine Learning, 27:259–286, 1997. [cytowanie na str. 8]

[162] Widmer G., Kubat M., *Effective learning in dynamic environments by explicit context tracking*, Proceedings of the European Conference on Machine Learning, 227–243, Springer-Verlag, London, UK, 1993. [cytowanie na str. 19]

[163] Widmer G., Kubat M., *Learning in the presence of concept drift hidden contexts*, Machine Learning, 23:69–101, 1996. [cytowanie na str. 8, 19, 90]

[164] Witten I. H., Frank E., *Data Mining. Practical Machine Learning Tools and Techniques*, Morgan Kaufmann Publishers, San Francisco, 2005. [cytowanie na str. 4, 5]

[165] World Health Organization. *Definition and diagnosis of diabetes mellitus and intermediate hyperglycemia*, Report of a WHO/IDF consultation, `http://whqlibdoc.who.int/publications/2006/9241594934_eng.pdf`. [cytowanie na str. 98, 99]

[166] Yang Q., Wu X., *10 challenging problems in data mining research*, International Journal of Information Technology and Decision Making, 5(4):597–604, 2006. [cytowanie na str. 20]

[167] Zięba M., Drwal M., Tomczak J. M. , Juszczyszyn K., *Typical motifs of e-mail based social networks*, (red.) L. Borzemski, w: Information Systems Architecture Technology: Designing, Development Implementation of Information Systems, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 127–133, 2008. [cytowanie na str. 1]

Spis symboli i skrótów

Symbol/skrót	Opis
$\mathbf{u} = [u^1\ u^2 \dots u^D]^{\mathrm{T}} \in \mathscr{U}$	wektor D wejść
$u^d \in \mathscr{U}_d$	d-te wejście
K_d	liczność zbioru wartości d-tego wejścia
K	liczność zbiorów wartości wejść
$y \in \mathscr{Y}$,	wyjście
Y	liczba wartości wyjściowych
$c_m \in \mathscr{C}$	kontekst
M	liczba zmian kontekstu
N	liczba obserwacji
N_m	liczba obserwacji dla m-tego kontekstu
$R(c_m)$ lub R_m	relacja
$\varphi(\mathbf{u}, y; c_m)$ lub $\varphi_m(\mathbf{u}, y)$	własność obiektu
$\varpi[\cdot]$	wartość logiczna
$\bar{R}$	relacja aproksymująca $R(c_m)$
Φ	model
L	długość okna
$\mathcal{D}$	ciąg uczący
$\mathcal{D}_m$	ciąg obserwacji dla kontekstu c_m
$\mathcal{D}_N^L$	obserwacje zawarte w oknie przesuwnym o długości L od momentu N
$\mathbf{u}_n$	n-ta obserwacja wejść
y_n	n-ta obserwacja wyjścia

Symbol/skrót	Opis
$\bar{y}_n$	n-te wyjście modelu
Q	kryterium jakości w przypadku deterministycznym
Q_p	kryterium jakości w przypadku losowym
$\overline{Q}$	predykcyjny błąd sekwencyjny
δ	metryka dyskretna (delta Kroneckera)
γ	współczynnik zapominania
$p(\cdot)$	rozkład prawdopodobieństwa
$\mathbb{E}[\cdot]$	wartość oczekiwana
$\varrho(\cdot,\cdot)$	miara niepodobieństwa
τ	zbiór momentów zmian kontekstu
σ	parametr wrażliwości
G_1, G_2, G_3	algorytm uczenia
AQ	algorytm AQ
$\mathrm{card}\{\cdot\}$	liczność zbioru
α_k^d	formuła wejściowa
α_l^{out}	formuła wyjściowa
$\wedge$	operator logiczny i
$\vee$	operator logiczny lub
$\Rightarrow$	operator logiczny *jeśli ... to ...*
ϕ	pojedyncza reguła
ϕ_{in}	warunek reguły
ϕ_{out}	decyzja reguły
$\mathscr{A}$	zbiór dopuszczalnych formuł elementarnych
VC-dim	wymiar Vapnika-Chervonenkisa
$\mathscr{P}$	przestrzeń rozkładów prawdopodobieństwa
$\pi(\cdot)$	prawdopodobieństwo a priori
$\boldsymbol{\theta}$	wektor parametrów
$\hat{\boldsymbol{\theta}}$	estymator największej wiarygodności parametrów
η	liczba ostatnio sprawdzanych klasyfikacji lub liczba obserwacji do oceny
ε	wartość progowa

Symbol/skrót	Opis
f	funkcja oceniająca
N_{ϕ}^{l}	liczba przykładów z klasy l pokrytych przez regułę ϕ
$\bar{N}$	liczba przykładów spoza klasy l
$\bar{N}_l$	liczba przykładów spoza klasy l pokrytych przez regułę
$\mathcal{G}$	graf
V	zbiór wierzchołków grafu G
E	zbiór łuków grafu G
$\mathcal{G}_0$, $\mathcal{G}_1$	graf negatywny i pozytywny
$e_{i,j}^{s \to t}$	łuk
$w_{l,i,j}^{s \to t}$	waga łuku
w_{l}^{out}	częstość występowania l-tego wyjścia
$\mathbf{w}_l$	wektor wag dla l-tego wyjścia
π	ścieżka w grafie
κ	liczba parametrów potrzebnych do zakodowania grafu
$\mathrm{code}(\cdot)$	kod 0-1 grafu
μ_c	miara pokrycia
μ_a	miara dokładności
$q(\cdot,\cdot)$	wartość kryterium syntetycznego
$q_{i,j}^{s \to t}$	wartość kryterium syntetycznego dla łuku $e_{i,j}^{s \to t}$
β	parametr ważący
$\mathbf{q}$	wektor wag determinujący graf dopuszczalnych rozwiązań
ϵ	wartość progowa
CNF	koniunkcyjna postać normalna
DNF	dysjunkcjyjna postać normalna
SOA	system o architekturze zorientowanej na usługi

Wytłuszczone symbole odnoszą się do wektorów i macierzy.

Spis rysunków

Spis tabel

Skorowidz

Incremental Knowledge Extraction from Data for Non-Stationary Objects

In this work a problem of knowledge extraction from data for non-stationary objects (phenomena) is considered. The knowledge is represented by decision rules. An object is described by a vector of input variables and an output variable. Additionally, it is assumed that the object is affected by an external quantity called *context*. The context can be viewed as all external factors that are unobservable and immeasurable, and which cause changes in the object's properties, e.g., an illness that influences patient's condition. Furthermore, it is assumed that the context changes in an abrupt manner. The problem of knowledge extraction is stated in two ways: i) objects are described by a deterministic properties, e.g., functional dependencies, ii) objects are described by joint probability distributions of input and output random variables. Nevertheless, in both cases the proposed algorithms are the same but the knowledge is interpreted differently. In the deterministic case, for fixed context, the object property is *discovered* from data. However, for the random description, the object is *imitated* so that to minimize the risk of decision making.

In order to follow changes of the object's properties an incremental learning is applied. Three approaches are used: i) *temporal batch learning*, ii) *learning with shifting window*, iii) *learning with self-adjusting model*. First approach is based on detecting moments of context's changes, i.e., dividing learning sequence into subsequences corresponding to different contexts. Second approach applies forgetting mechanism with shifting window, and the third one – forgetting mechanism with forgetting factor (exponential forgetting).

According to the first approach two algorithms are proposed. The first one applies *frequentist* modeling to context change detection which is based on comparing two probability

distributions using a dissimilarity measure, e.g., Bhattacharyya measure, Kullback-Leibler measure. The second algorithm adapts *Bayesian* modeling which allows to compare two models (one corresponds to no context change and second - one context change) via *Bayes factor*. However, usually the evaluation of the Bayes factor is analytically intractable and that is why its approximation is calculated.

Furthermore, two algorithms for rules induction with shifting window are presented. Both of them are modifications of the well-known *Algorithm Quasi-optimal* (AQ). The first method uses the AQ for decision rules extraction basing on observations maintained in the shifting window. The second algorithm applies additional implicit forgetting mechanism which removes rules according to the given *rule evaluation function*.

Moreover, an algorithm with self-adjusting model is formulated. The main idea of the algorithm is to use graph representation to aggregate observations and induce rules. The application of graphs introduces a new way of regularization that limits the search space. Besides, the knowledge is accumulated in the weights of the graphs and therefore the exponential forgetting is easily applied.

At the end of the work the proposed algorithms are evaluated in empirical studies. The performance of context change detection methods are verified in the two examples. The first task concerns the real-life dataset about changes in the time series of fatal disasters in coal mines in the United Kingdom since 1851 to 1962. The second one is about detecting changes in the performance index of an execution system of a service-oriented system. The performance of the rules induction algorithms are assessed in the three examples. The first example is based on *STAGGER* dataset which is about a deterministic object and two changes of the context. The second one concerns the real-life dataset, called *Electricity*, which is about changes in the electricity price in Australia. The last task is connected with the knowledge extraction to support diabetes treatment. The illustrating examples show the advisability of presented algorithms and reveal a wide spectrum of the existing methodological problems.