POLITECHNIKA BIAŁOSTOCKA

WYDZIAŁ INFORMATYKI

KATEDRA MEDIÓW CYFROWYCH I GRAFIKI KOMPUTEROWEJ

STRESZCZENIE ROZPRAWY DOKTORSKIEJ

ROZMYTE LASY LOSOWE OPARTE NA MODELACH KLASTROWYCH DRZEW DECYZYJNYCH W ZADANIACH KLASYFIKACJI

WYKONAWCA: MGR INŻ. ŁUKASZ GADOMER

PROMOTOR: DR HAB. ZENON A. SOSNOWSKI, PROF. PB

BIAŁYSTOK 2018 r.

Spis treści

1. Wstęp

1.1 Zagadnienia metodologiczne

1.1.1 Problem badawczy

Głównym problemem badawczym, z którym zmierzyłem się w ramach niniejszej rozprawy, jest problem klasyfikacji danych. Dodatkowo, zająłem się również problemem regresji danych.

1.1.2 Przedmiot badań

Grupa metod klasyfikacji oraz, dodatkowo, regresji danych opracowana w ramach prowadzonych przeze mnie prac badawczych – rozmyte lasy losowy oparte na modelach klastrowych drzew decyzyjnych – jest przedmiotem niniejszej rozprawy.

1.1.3 Cel rozprawy

Celem rozprawy jest opracowanie metody służącej do klasyfikacji oraz, dodatkowo, regresji danych, stanowiącej klasyfikator zbiorczy, opary o klastrowe–rozmyte drzewa decyzyjne oraz klastrowo–kontekstowe rozmyte drzewa decyzyjne, wykorzystujący zagadnienie losowości. Metoda ta powinna być dostosowana do radzenia sobie z wieloma rodzajami problemów. Stworzony klasyfikator powinien osiągać wyniki lepsze pod względem jakości klasyfikacji (czas ma znaczenie drugorzędne) niż inne powszechnie znane klasyfikatory z dziedziny klasyfikatorów opartych na drzewach decyzyjnych. Metoda powinna posiadać szereg udoskonaleń, które mogą być wykorzystane (lub nie – w zależności od zadania) w ramach danego problemu klasyfikacji (lub, dodatkowo, regresji). Opracowana metoda rozmytych lasów losowych opartych na modelach klastrowych drzew decyzyjnych powinna osiągać konkurencyjne wyniki (w stosunku do klasyfikatorów stanowiących bazę porównania) w zadaniach klasyfikacji. Dodatkowym atutem będzie, jeśli stworzone rozwiązanie osiągnie atrakcyjne, w porównaniu do innych rozwiązań, rezultaty w zadaniach regresji.

1.1.4 Hipotezy badawcze

Tworząc omawiany w ramach niniejszej rozprawy klasyfikator założyłem, że spełni on oczekiwania wymienione w poprzednim podrozdziale. Opracowana w ramach pracy metoda łącząca w sobie zalety klasyfikatora zbiorczego, zagadnienia losowości oraz klastrowych–rozmytych drzew decyzyjnych lub klastrowo–kontekstowych rozmytych drzew decyzyjnych, powinna pozwalać na osiągnięcie satysfakcjonujących wyników klasyfikacji oraz – dodatkowo – regresji danych. Poprzez „satysfakcjonujące wyniki" rozumie się, że osiągane jakości klasyfikacji / regresji powinny być lepsze, niż rezultaty otrzymywane przy użyciu innych klasyfikatorów z dziedziny drzew (lub przynajmniej porównywalne), a w szczególności lepsze, niż wyniki klasyfikacji osiągane przez drzewa wykorzystane w ramach tego klasyfikatora działające samodzielnie i bez dodatkowych udoskonaleń. Dodatkowo, powinna istnieć możliwość wykorzystania opracowanej metody do szerokiego spektrum problemów. Tak sformułowane hipotezy badawcze zostaną w ramach niniejszej rozprawy zweryfikowane.

2. Rozmyty–klastrowy las losowy

Rozmyty–klastrowy las losowy to klasyfikator, który został opracowany w ramach niniejszej rozprawy. W rozdziale tym przedstawiono jego koncepcję, ideę i sposób działania.

2.1 Koncepcja rozmytego–klastrowego lasu losowego

Rozmyty–klastrowy las losowy jest nowym klasyfikatorem stworzonym w ramach niniejszej rozprawy. Istota tego rozwiązania zakłada połączenie dwóch klasyfikatorów: rozmytego lasu losowego [9] oraz jednego z dwóch rodzajów rozmytych drzew decyzyjnych (w zależności od wariantu): klastrowych–rozmytych drzew decyzyjnych [81] lub klastrowo–kontekstowych rozmytych drzew decyzyjnych [94]. W swoich założeniach stworzony klasyfikator zbiorczy powinien działać podobnie do rozmytego lasu losowego, ale zamiast tradycyjnych rozmytych drzew decyzyjnych wykorzystywać jeden z dwóch rodzajów wspomnianych drzew. Rozmyty las losowy wykorzystuje siłę losowości w celu poprawy jakości klasyfikacji. Klastrowe–rozmyte drzewa decyzyjne oraz klastrowo–kontekstowe rozmyte drzewa decyzyjne są tworzone losowo niejako z definicji, jako że środki ich klastrów (stanowiące macierz przynależności) są wybierane losowo na początku algorytmu tworzenia drzewa. Połączenie tych dwóch klasyfikatorów, tak aby ich mocne strony działały ze sobą w synergii, powinno skutkować otrzymaniem obiecujących wyników.

2.2 Rozmyty–klastrowy las losowy z klastrowymi–rozmytymi drzewami decyzyjnymi

Pierwszą z zaproponowanych wersji rozmytego–klastrowego lasu losowego jest rozmyty–klastrowy las losowy z klastrowymi–rozmytymi drzewami decyzyjnymi. Drzewami, które wchodzą w skład tego rozwiązania, są klastrowe–rozmyte drzewa decyzyjne. Natura tego klasyfikatora pozwala oczekiwać, że będzie on szczególnie dobrze radził sobie z zadaniami klasyfikacji danych. Ze względu na długą nazwę, w tabelach oraz spisach w niniejszej pracy ta wersja klasyfikatora będzie nazywana wersją 1 bądź C–FF (bez wykorzystania losowości, ang. *Cluster–Fuzzy Forest*) lub C–FRF (z wykorzytstaniem losowości, ang. *Cluster–Fuzzy Random Forest*).

Utworzony rozmyty–klastrowy las losowy może być wykorzystany w procesach klasyfikacji oraz regresji. Klasyfikacja odbywa się na zasadzie głosowania większościowego na podstawie decyzji klastrowych–rozmytych drzew decyzyjnych wchodzących w skład klasyfikatora zbiorczego. Wynik regresji obliczany jest poprzez uśrednienie wyników uzyskanych przez wszystkie drzewa w lesie.

Rozmyty–klastrowy las losowy z klastrowymi–rozmytymi drzewami decyzyjnymi został w ramach niniejszej rozprawy przetestowany w zadaniach klasyfikacji.

2.3 Rozmyty–klastrowy las losowy z klastrowo–kontekstowymi rozmytymi drzewami decyzyjnymi

Druga zaproponowana wersja rozmytego–klastrowego lasu losowego wykorzystuje klastrowo–kontekstowe rozmyte drzewa decyzyjne. Podobnie jak w przypadku wersji wykorzystującej klastrowe–rozmyte drzewa decyzyjne, klasyfikator spełnia wszystkie założenia opisane w rozdziale 2. Wykorzystanie kontekstów dzielących atrybut decyzyjny pozwala wnioskować, że ten rodzaj lasu szczególnie dobrze radził sobie będzie z problemami regresji. Ze względu na długą nazwę, w tabelach oraz spisach w niniejszej pracy wersja ta będzie nazywana wersją 2 bądź CC–FF (bez wykorzystania losowości, ang. *Cluster–Context Fuzzy Forest*) lub CC–FRF (z wykorzytstaniem losowości, ang. *Cluster–Context Fuzzy Random Forest*).

Po zbudowaniu, rozmyty–klastrowy las losowy może zostać użyty do klasyfikacji oraz regresji danych. Rezultatem klasyfikacji jest wynik głosowania większościowego dokonanego przez klastrowo–kontekstowe rozmyte drzewa decyzyjne wchodzące w skład lasu. Rezultatem regresji jest średnia wartość wyników otrzymanych przez poszczególne drzewa wchodzące w skład lasu.

W ramach niniejszej rozprawy, sprawdzono działanie rozmytego–klastrowego lasu losowy z klastrowo–kontekstowymi rozmytymi drzewami decyzyjnymi w zadaniach regresji.

3. Ważone podejmowanie decyzji w rozmytym–klastrowym lesie losowym

Wiedza przechowywana w klasyfikatorze będącym przedmiotem badań w ramach niniejszej rozprawy ma charakter rozmyty. Obiekty przynależą równocześnie do różnych węzłów (klas decyzyjnych) w różnym stopniu. Drzewa wchodzące w skład lasu także mogą być lepsze lub gorsze pod względem zdolności predykcyjnej. Zarówno w przypadku przynależności obiektów do węzłów, jak i podejmowaniu decyzji przez las korzystając z decyzji drzew, warto rozważyć ważenie decyzji – tak, aby niektóre węzły oraz niektóre drzewa miały większy wpływ na decyzję od innych. W tym podrozdziale przedstawione zostały wybrane sposoby ważenia decyzji, z wyszczególnieniem tych, które zostały użyte w niniejszej pracy.

3.1 Wykorzystanie operatorów OWA w rozmytym–klastrowym lesie losowym

W celu poprawy jakości klasyfikacji rozmytego–klastrowego lasu losowego z klastrowymi–rozmytymi drzewami decyzyjnymi zgodnie z opisanymi założeniami wykorzystane zostały operatory uporządkowanego ważonego uśredniania. Zaproponowano trzy różne warianty użycia operatorów OWA w opracowanej metodzie:

- Lokalne OWA – wagi obliczane dla każdego węzła każdego drzewa w lesie,
- Globalne OWA:
  - Globalne OWA dla każdego drzewa w lesie – wagi obliczane dla każdego drzewa w lesie,
  - Globalne OWA dla lasu – wagi obliczane dla całego lasu.

3.1.1 Rodzaje OWA

W niniejszym rozdziale zaprezentowane zostały trzy sposoby wykorzystania operatorów uporządkowanego ważonego uśredniania: lokalne OWA, globalne OWA dla drzewa oraz globalne OWA dla lasu.

3.1.1.1 Lokalne OWA

Pierwszym z zaproponowanych sposobów wykorzystania operatorów OWA w rozmytym–klastrowym lesie losowym jest lokalne OWA. Koncepcja ta zakłada, że każda decyzja podejmowana w rozmytym–klastrowym lesie losowym, zarówno w ramach drzewa, jak i pojedynczego węzła, jest ważona przy użyciu operatora uporządkowanego ważonego uśredniania. Aby realizacja tej idei była możliwa, do każdego węzła w każdym drzewie w ramach klasyfikatora zbiorczego przypisana jest macierz wag operatorów OWA. Oba wymiary tej macierzy są równe liczbie węzłów potomnych każdego węzła (założonej liczby klastrów c, na jaką dzielony jest każdy węzeł w drzewie podlegający podziałowi). Informacja o stopniach przynależności obiektów ze zbioru uczącego do każdego węzła jest znana i zapamiętywana. Następnie, jest ona używana do obliczenia wag operatorów OWA dla każdego węzła. Do obliczenia wag operatorów OWA odpowiadających danemu potomkowi węzła wykorzystywane są obiekty ze zbioru uczącego, które mają najwyższy stopień przynależności odpowiadający temu właśnie potomkowi.

Optymalne wagi operatorów lokalnego OWA dla węzła p oraz klastra k obliczane są poprzez maksymalizację poniższej funkcji:

$$OWA(\boldsymbol{W}_{p_k}) = \sum_{j=1}^{c} \sum_{i=1}^{|M_j|} \boldsymbol{U}_{ij} \boldsymbol{W}_{p_{kj}} \tag{3.1}$$

Po utworzeniu wszystkich drzew i wyznaczeniu wszystkich operatorów uporządkowanego ważonego uśredniania rozmyty–klastrowy las losowy może zostać wykorzystany do klasyfikacji danych. W wariancie z lokalnym OWA, każde drzewo wchodzące w skład lasu, zamiast wybierać maksymalnego stopnia przynależności obiektu do węzła:

$$U_{max} = \max_{i \in 1,2,\ldots,|M|, j \in 1,2,\ldots,c} \boldsymbol{U}_{ij} \tag{3.2}$$

wybiera maksymalny rezultat mnożenia stopnia przynależności obiektu do węzła przez odpowiadającą mu wagę operatora OWA, pobraną z węzła:

$$k = arg \max_{i \in 1,2,\ldots,|M|, j \in 1,2,\ldots,c} j \, \boldsymbol{U}_{ij} \tag{3.3}$$

$$U_{max} = \max_{i \in 1,2,\ldots,|M|, j \in 1,2,\ldots,c} \boldsymbol{U}_{ij} \times \boldsymbol{W}_{kj} \tag{3.4}$$

Po przeprowadzeniu klasyfikacji przez poszczególne drzewa ostateczna decyzja o przynależności obiektu podejmowana jest na podstawie głosowania większościowego.

3.1.1.2 Globalne OWA dla drzewa

W przypadku globalnego OWA dla drzewa, wagi obliczane są dla każdego drzewa wchodzącego w skład rozmytego–klastrowego lasu losowego po utworzeniu drzewa. Zasady tworzenia operatorów uporządkowanego ważonego uśredniania są takie same, jak w przypadku lokalnego OWA.

Optymalne wagi operatorów OWA dla drzewa t oraz klastra k obliczane są poprzez maksymalizację funkcji:

$$OWA(\boldsymbol{W}_{t_k}) = \sum_{j=1}^{c}\sum_{i=1}^{N}\boldsymbol{U}_{ij}\boldsymbol{W}_{t_{kj}} \tag{3.5}$$

Utworzony las może zostać wykorzystany do klasyfikacji danych. W ramach tego procesu dla każdego klasyfikowanego przez drzewo obiektu o indeksie i, zamiast wybierania maksymalnego stopnia przynależności obiektu do jednego z węzłów, tak jak byłoby to dokonane w ramach głosowania większościowego:

$$U_{i_{max}} = \max_{j \in 1,2,\ldots,c} \boldsymbol{U}_{ij} \tag{3.6}$$

wybierany jest maksymalny wynik mnożenia stopnia przynależności przez odpowiadającą mu wagę pobraną z operatora OWA dla drzewa t:

$$k = arg \max_{j \in 1,2,\ldots,c} j\,\boldsymbol{U}_{ij} \tag{3.7}$$

$$U_{i_{max}} = \max_{j \in 1,2,\ldots,c} \boldsymbol{U}_{ij} \times \boldsymbol{W}_{t_{kj}} \tag{3.8}$$

3.1.1.3 Globalne OWA dla lasu

Ostatni spośród zaproponowanych typów uporządkowanego ważonego uśrednienia nazwany został globalnym OWA dla lasu. W ramach tej metody dla całego lasu obliczany jest tylko jeden zestaw wag operatorów OWA. Decyzja podejmowana przez każdy węzeł w każdym drzewie w lesie ważona jest tymi właśnie wagami, współdzielonymi między drzewami i węzłami.

Optymalne wagi operatorów OWA dla klastra k obliczane są poprzez maksymalizację funkcji:

$$OWA(\boldsymbol{W}_k) = \sum_{j=1}^{c}\sum_{l=1}^{T}\sum_{i=1}^{N}\boldsymbol{U}_{l_{ij}}\boldsymbol{W}_{kj} \tag{3.9}$$

Kiedy las oraz operator uporządkowanego ważonego uśredniania są gotowe, mogą zostać użyte w procesie klasyfikacji. Podczas przeprowadzania tego procesu, zamiast wybierać maksymalny stopień przynależności obiektu o indeksie i do węzła, jak odbywałoby się to bez użycia wag:

$$U_{i_{max}} = \max_{j\in 1,2,...,c}\boldsymbol{U}_{ij} \tag{3.10}$$

wybierany jest maksymalny wynik mnożenia stopnia przynależności oraz odpowiadającej mu wagi z operatora OWA dla całego lasu:

$$k = arg\max_{j\in 1,2,...,c} j\,\boldsymbol{U}_{ij} \tag{3.11}$$

$$U_{i_{max}} = \max_{j\in 1,2,...,c}\boldsymbol{U}_{ij}\times\boldsymbol{W}_{kj} \tag{3.12}$$

3.2 Ważenie proporcjonalne w rozmytym–klastrowym lesie losowym

Sposób ważonego podejmowania decyzji przez rozmyty–klastrowy las losowy z klastrowo–kontekstowymi rozmytymi drzewami decyzyjnymi, nazwany ważeniem proporcjonalnym, został opracowany i zaprezentowany w ramach [40]. Zgodnie z nim, największy wpływ na decyzję w lesie ma drzewo, które w procesie nauki lasu osiągnęło najlepsze rezultaty, najmniejszy zaś te, które pozwoliło na otrzymanie w procesie nauki rezultatów najgorszych. Pozostałe drzewa posiadają wagi umieszczone pomiędzy tymi dwoma skrajnościami.

Niech $W = [w_1, w_2, ..., w_T]$ będzie wektorem wag, gdzie $w_t \in [0,T]$, $\sum_{t=1}^{T} w_t = \dfrac{T(T+1)}{2}$, $Pred_{kt}$ decyzją drzewa t dotyczącą klasyfikacji obiektu k, a $Pred_k$ jest ostateczną decyzją lasu o klasyfikacji obiektu k. Ważenie decyzji dotyczącej obiektu k podejmowanej przez las odbywa się w następujący sposób:

$$Pred_k = \frac{\sum_{t=1}^{T} \left(Pred_{kt} \times w_t \right)}{\sum_{t=1}^{T} w_t} \tag{3.13}$$

Zaproponowano dwa sposoby określenia stopnia jakości drzew (a co za tym idzie, jego wpływu na ostateczną decyzję) w rozmytym–klastrowym lesie losowym:

- na podstawie obiektów ze zbioru uczącego,
- na podstawie indeksu wydajności drzewa.

Skuteczność działania opracowanej metody ważonego podejmowania decyzji sprawdzono eksperymentalnie. Wyniki eksperymentów zamieszczone zostały w rozdziale **??**.

3.2.1 Ważenie proporcjonalne na podstawie zbioru uczącego

Obliczanie stopnia jakości drzew na podstawie obiektów ze zbioru uczącego odbywa się w następujący sposób. Na początku tworzony jest wektor wag W. Początkowo wektor ten wypełniany jest wartościami $\frac{T}{2}$. Obliczany jest krok $step$, o jaki modyfikowane będą wagi w ramach każdej kolejnej iteracji. Jest on dobierany tak, aby najmniejsza waga nie spadła poniżej 0, a największa nie przekroczyła T. Następnie przeprowadzana jest liczba iteracji równa liczności zbioru uczącego. W ramach każdej iteracji obiekt z tego zbioru jest poddawany procesowi regresji. Waga każdego drzewa jest modyfikowana w zależności od tego, jak dobrze dane drzewo poradziło sobie z przewidzeniem wartości atrybutu decyzyjnego tego obiektu – jest ona proporcjonalnie do rezultatu podnoszona lub obniżana. Po zakończeniu procesu powstaje wektor wag, który służy do określenia stopnia udziału drzewa na ostateczną decyzję.

3.2.2 Ważenie proporcjonalne na podstawie indeksu wydajności drzewa

Drugim zaproponowanym sposobem ważenia obiektów w rozmytym–klastrowym lesie losowym jest wykorzystanie indeksu wydajności drzewa. Dla dowolnego drzewa t, dla problemów o ciągłym atrybucie decyzyjnym, indeks wydajności drzewa ef_t można obliczyć w następujący sposób:

$$ef_t = \frac{\sum_{i=1}^{N} |y_t - \bar{y}|}{N} \tag{3.14}$$

gdzie $\bar{y}$ jest oczekiwaną wartością atrybutu decyzyjnego, a y_t jest wartością atrybutu decyzyjnego otrzymaną przez drzewo t.

Choć w ramach eksperymentów nad skutecznością opracowanej metody ważenie proporcjonalne wykorzystano jedynie do rozmytych–klastrowych lasów losowych z klastrowo–kontekstowymi rozmytymi drzewami decyzyjnymi oraz do problemów o ciągłym atrybucie decyzyjnym, ważenie przy użyciu indeksu wydajności drzewa można także zastosować do wersji klasyfikatora z klastrowymi–rozmytymi drzewami decyzyjnymi i problemów klasyfikacji. W takim przypadku indeks wydajności drzewa obliczony będzie następująco:

$$ef_t = \frac{N_{corr_t}}{N} \tag{3.15}$$

gdzie N_{corr_t} jest liczbą poprawnie sklasyfikowanych obiektów ze zbioru uczącego przez drzewo t.

Po obliczeniu indeksów wydajności wszystkich drzew poszczególnym drzewom przypisywane są wagi będące liczbami całkowitymi od 1 do T zgodnie z rosnącymi indeksami wydajności (drzewo o najmniejszym indeksie wydajności otrzymuje wagę 1, drzewo o największym indeksie wydajności otrzymuje wagę T).

4. Przycinanie drzew w rozmytym–klastrowym lesie losowym

W niniejszym rozdziale zaprezentowane zostały sposoby przycinania klastrowych–rozmytych drzew decyzyjnych oraz klastrowo–kontekstowych rozmytych drzew decyzyjnych w rozmytym–klastrowym lesie losowym.

4.1 Przycinanie drzewa w trakcie wzrostu

Koncepcja przycinania drzewa w trakcie wzrostu opiera się na zatrzymaniu procesu tworzenia drzewa w pewnych konkretnych sytuacjach w celu uniknięcia jego nadmiernego rozrostu. Idea ta jest użyta w klastrowych–rozmytych drzewach decyzyjnych oraz klastrowo–kontekstowych rozmytych drzewach decyzyjnych, wchodzących w skład klastrowego–rozmytego lasu losowego, niejako z definicji. Wymienione drzewa mają następujące kryteria zatrzymania wzrostu drzewa:

- wszystkie węzły osiągnęły niższy stopień zróżnicowania niż założona wartość graniczna,
- w żadnym węźle nie pozostało wystarczająco dużo obiektów, aby wykonać podział,
- indeks strukturalności drzewa (wartość symbolizująca zróżnicowanie całego drzewa) osiągnął wartość mniejszą niż założona wartość graniczna,
- liczba wykonanych iteracji algorytmu (podziałów węzłów) osiągnęła założoną wartość graniczną.

Kryteria te stanowią realizację idei przycinania klastrowych–rozmytych drzewach decyzyjnych oraz klastrowo–kontekstowych rozmytych drzewach decyzyjnych w trakcie ich wzrostu. Wszystkie tworzone w ramach prac nad niniejszą rozprawą drzewa tworzone były z wykorzystaniem tych kryteriów zatrzymania, realizują więc one koncepcję przycinania w trakcie wzrostu z definicji.

4.2 Przycinanie rozrośniętego drzewa

W niniejszej rozprawie nacisk położony został na metodach przycinania rozrośniętego drzewa, jako że te właśnie metody zostały bezpośrednio zastosowane w stworzonym klasyfikatorze. Powody takiej decyzji zostały wyjaśnione w rozdziale 4.1. Istnieje wiele metod przycinania rozrośniętych drzew. Powstały liczne prace, w których porównano skuteczność ich działania oraz inne aspekty z nimi związane, na przykład [31]. Podobne porównania przeprowadzone zostały także na rozmytych drzewach decyzyjnych, na przykład w [85]. Spośród wielu znanych metod przycinania, do opracowanych w ramach niniejszej rozprawy rozwiązań wybrano, dostosowano i zaimplementowane następujące:

- Przycinanie oparte na kryterium redukcji błędu (ang. *Reduced Error Pruning – REP*),
- Przycinanie oparte na kryterium pesymistycznego błędu (ang. *Pessimistic Error Pruning – PEP*),
- Przycinanie oparte na kryterium minimalnego błędu (ang. *Minimum Error Pruning – MEP*)
- Przycinanie oparte na kryterium wartości krytycznej (ang. *Critical Value Pruning – CVP*),
- Przycinanie oparte na kryterium kosztu–złożoności (ang. *Cost–Complexity Pruning – CCP*).

4.2.1 Przycinanie oparte na kryterium redukcji błędu

Jedną z najprostszych, najbardziej zrozumiałych i najpopularniejszych metod przycinania rozrośniętego drzewa jest przycinanie oparte na kryterium redukcji błędu (ang. *Reduced Error Pruning – REP*). Metoda ta została zaproponowana przez Quinlana w [83]. Idea tego algorytmu przycinania opiera się na zastępowaniu każdego węzła, który posiada węzły potomne (przeszukując drzewo od dołu do góry), liściem i porównywanie jakości klasyfikacji. Jeśli wynik osiągnięty przy pomocy zastąpionego węzła jest lepszy lub równy wynikowi osiągniętemu przez oryginalny węzeł, drzewo pozostaje przycięte, a algorytm kontynuuje sprawdzanie kolejnych węzłów. W przeciwnym razie, jeśli liczba niepoprawnie sklasyfikowanych obiektów przed przycięciem gb_p będzie mniejsza, niż liczba niepoprawnie sklasyfikowanych obiektów po przycięciu g_p: $gb_p < g_p$, przywracane jest oryginalne poddrzewo o korzeniu w węźle, który przycięto. W rezultacie otrzymywane jest najmniejsze drzewo, którego jakość klasyfikacji / regresji jest taka sama lub lepsza, jak w przypadku drzewa oryginalnego.

Poza zrozumiałością i prostotą, główną zaletą tej metody jest jej liniowa złożoność obliczeniowa. Każdy węzeł jest odwiedzany tylko raz, co skutkuje relatywnie szybkim działaniem algorytmu, w szczególności w porównaniu do konkurencyjnych sposobów. Metoda ta wymaga oddzielnego zbioru przycinania, co może być kłopotliwe w przypadku małych zbiorów danych. Pomijając naiwność metody, przycinanie oparte na kryterium redukcji błędu, powołując się na różne badania, może być użytecznym algorytmem przycinania rozrośniętych drzew i często pozwala na osiągnięcie lepszych wyników, niż inne, bardziej skomplikowane metody [31].

4.2.2 Przycinanie oparte na kryterium pesymistycznego błędu

Kolejną metodą przycinania rozrośniętych drzew zaproponowaną przez Quinlana w [83] jest przycinanie oparte na kryterium pesymistycznego błędu (ang. *Pessimistic Error Pruning – PEP*). Bazuje ona na obserwacji, że błędy klasyfikacji osiągnięte przez drzewo używające zbioru uczącego są przesadnie optymistyczne, co prowadzi do nadmiernego rozrostu drzewa, gdy ten zbiór jest używany do przycinania. W celu osiągnięcia bardziej realistycznego oszacowania stopnia błędnych klasyfikacji, Quinlan zaproponował skorygowanie osiągniętego wyniku w sposób opisany w kolejnych akapitach.

Na początku obliczana jest g'_p – liczba niepooprawnie sklasyfikowanych obiektów po przycięciu podwęzłów węzła p, skorygowana o stałą wartość 0,5 ([83]):

$$g'_p = g_p + 0,5 \tag{4.1}$$

Następnie należy obliczyć g'_{T_p} – liczbę niepoprawnie sklasyfikowanych obiektów przez poddrzewo o korzeniu p przed przycięciem, skorygowaną o wartość określoną przez liczbę podwęzłów p:

$$g'_{T_p} = \sum g_l + \frac{P_{T_p}}{2} \tag{4.2}$$

gdzie T_p jest poddrzewem o korzeniu p przed przycięciem, P_{T_p} liczbą liści nieprzyciętego poddrzewa T_p, g_l liczbą niepoprawnie sklasyfikowanych obiektów ze zbioru uczącego przez liść l, a $\sum g_l$ liczbą niepoprawnie sklasyfikowanych obiektów ze zbioru uczącego przez poddrzewo o korzeniu p przed przycięciem.

Kolejnym krokiem jest obliczenie $SE(g'_{T_p})$ – standardowego błędu dla liczby niepoprawnie sklasyfikowanych obiektów przez poddrzewo o korzeniu p przed przycięciem ([83]):

$$SE(g'_{T_p}) = \sqrt{\frac{g'_{T_p} \times (G_p - g'_{T_p})}{G_p}} \tag{4.3}$$

gdzie G_p jest liczbą obiektów ze zbioru uczącego w węźle p.

Przycinanie węzła p odbywa się, gdy $g'_p > g'_{T_p} + SE(g'_{T_p})$. Oznacza to, że rozmiar drzewa jest zmniejszany gdy skorygowana liczba błędnych klasyfikacji osiągnięta przez przycięte drzewo jest większa niż skorygowany błąd przed przycięciem, powiększony o błąd standardowy.

Drzewo przeszukiwane jest od góry do dołu, każdy węzeł jest więc odwiedzany co najwyżej raz. Sprawia to, że złożoność obliczeniowa algorytmu jest, w najgorszym przypadku, liniowa. Wykonywane obliczenia są względnie mało skomplikowane, co sprawia, że algorytm działa szybko, nawet w porównaniu do innych metod przycinania uważanych za szybkie. Co więcej, przycinanie oparte na kryterium pesymistycznego błędu nie wymaga zbioru przycinania, co stanowi kolejną zaletę tego sposobu.

Z drugiej strony, statystyczne uzasadnienie tego sposobu przycinania budzi wątpliwości. Stała wartość użyta do korekty błędu klasyfikacji została dobrana heurystycznie i nie jest pewne, czy pasuje ona do wszystkich rodzajów problemów. Niezależnie od tego, liczne badania pokazują, że przycinanie oparte na kryterium pesymistycznego błędu często pozwala na osiągnięcie dobrych rezultatów, co w połączeniu z niepodważalnymi zaletami czyni tę metodę atrakcyjną opcją przycinania rozrośniętego drzewa.

4.2.3 Przycinanie oparte na kryterium minimalnego błędu

Niblett i Bratko w [76] zaproponowali metodę przycinania rozrośniętych drzew nazwaną przycinaniem opartym na kryterium minimalnego błędu (ang. *Minimum Error Pruning – MEP*). Do wyjaśnienia działania metody niech posłużą poniższe wzory.

Współczynnik błędu $Err(p)$ obliczany jest w następujący sposób:

$$Err(p) = \frac{G_p - G_{ps_{max}} + S - 1}{G_p + S} \tag{4.4}$$

gdzie S jest liczbą klas, s_{max} klasą reprezentowaną przez największą liczbę obiektów spośród wszystkich klas w ramach węzła p, a $G_{ps_{max}}$ liczbą obiektów ze zbioru uczącego przypisanych do klasy s_{max} w węźle p,

Ważony współczynnik błędu $WErr(p)$ wszystkich węzłów potomnych p obliczany jest zgodnie z następującym wzorem:

$$WErr(p) = \sum_{i=1}^{P_{T_p}} \frac{G_i}{G_p} \frac{G_i - G_{is_{max}} + S - 1}{G_i + S}, \sum G_i = G_p \tag{4.5}$$

Dla każdego wewnętrznego węzła p (przeszukiwanie odbywa się od dołu do góry) obliczany jest oczekiwany współczynnik błędu zgodnie z przytoczonymi wzorami. Następnie jest on porównywany z sumą ważonych współczynników błędów wszystkich węzłów potomnych p w oryginalnym drzewie. Współczynniki obliczane są w ten sam sposób (na podstawie zaprezentowanego wzoru), natomiast ważenie odbywa się na podstawie liczby obiektów w każdym z węzłów potomnych (która to liczba może zostać sprawdzona, przeszukując poddrzewo od dołu do góry). Jeśli ważony współczynnik błędu nieprzyciętego drzewa składającego się z węzłów potomnych węzła p jest większy lub równy niż współczynnik błędu węzła p po przycięciu drzewa ($WErr(p) \geq Err(p)$), drzewo jest przycinane (węzły potomne węzła p zostają usunięte).

Metoda przycinania na podstawie kryterium minimalnego błędu nie wymaga zbioru przycinania. Każdy węzeł jest odwiedzony raz, ale dla każdego węzła konieczne jest przejście po poddrzewie o korzeniu stanowionym przez ten węzeł. Sprawia to, że przycinanie oparte na kryterium minimalnego błędu jest wolniejsze niż algorytmy o liniowej złożoności obliczeniowej. To, jak bardzo drzewo zostaje przycięte, silnie zależy od liczby klas w zbiorze danych, co może prowadzić do niestabilnych wyników. Co więcej, najlepsze wyniki osiągane przez drzewo przycinane tym algorytmem mogą być uzyskane, gdy każda klasa ze zbioru danych reprezentowana jest przez podobną liczbę obiektów, co zdarza się raczej rzadko.

4.2.4 Przycinanie oparte na kryterium wartości krytycznej

Mingers zaproponował algorytm przycinania opartego na kryterium wartości krytycznej w (ang. *Critical Value Pruning – CVP*) [71]. Metoda ta, chociaż przycina drzewo po jego utworzeniu, podobna jest do metod przycinania drzewa w trakcie wzrostu. Głównym założeniem tego sposobu przycinania jest wykorzystanie wartości granicznej do oceny stopnia ważności węzła. Jeśli wartość krytyczna węzła mieści się poniżej założonej wartości granicznej, drzewo jest przycinane w tym węźle poprzez usunięcie jego węzłów potomnych. Jeśli jednak rozważany węzeł nie przekroczy wartości granicznej, ale którykolwiek z jego węzłów potomnych osiągnie tę wartość, drzewo nie jest przycinane w miejscu rozważanego węzła. W przypadku przekroczenia przez węzeł wartości granicznej przycięcie nie następuje. Drzewo przeszukiwane jest od dołu do góry.

Wartość graniczna decyduje o stopniu przycinania drzewa. Drzewo zostaje przycięte tym bardziej, im większa wartość graniczna została wybrana. Autor zaproponował, aby wartość ta była iteracyjnie zwiększana, a drzewo powstałe w wyniku przycinania w ramach każdej iteracji zostało zapisywane. Następnie, po zakończeniu iteracyjnego zwiększania wartości granicznej, ze zbioru zapisanych drzew wybierane jest najlepsze z nich (to znaczy takie, które osiągnęło największą skuteczność klasyfikacji). Wybrane w ten sposób drzewo stanowi ostateczny wynik przycinania za pomocą opisanej metody.

W klastrowych–rozmytych drzewach decyzyjnych oraz klastrowo–kontekstowych rozmytych drzewach decyzyjnych istnieje potrzeba ustalenia parametru, który pełniłby rolę wartości granicznej. Zdecydowano, że posłuży do tego zróżnicowanie węzła. Wybór tego parametru jako wartości granicznej jest uzasadniony, ponieważ w procesie tworzenia klastrowych–rozmytych drzew decyzyjnych oraz klastrowo–kontekstowych rozmytych drzew decyzyjnych zróżnicowanie węzła decyduje o tym, czy węzeł zostanie podzielony. Logiczne jest więc przyjęcie tego samego parametru jako decydującego o ewentualnym przycięciu drzewa w ramach danego węzła.

Wartość graniczna zróżnicowania węzła V_i – parametru, od którego uzależniana jest decyzja o przycięciu drzewa w danym węźle – zmienia się w kolejnych iteracjach algorytmu przycinania opartego na kryterium wartości krytycznej. W ramach każdej iteracji zależy ona od założonej wartości minimalnej zróżnicowania V_{min}, założonej wartości maksymalnej zróżnicowania V_{max} oraz liczby iteracji (a więc liczby przyciętych drzew, z których zostanie wybrane jedno ostateczne jako rezultat przycinania) $|T_{prun}|$ ($\boldsymbol{T}_{prun}$ jest zbiorem przyciętych drzew, spośród których wybierane jest najlepsze jako finalny rezultat przycinania). Dla i-tej iteracji wartość graniczna zróżnicowania jest ona następująco:

$$V_i = V_{min} + i \times \frac{V_{max} - V_{min}}{|T_{prun}|} \tag{4.6}$$

Przycinanie oparte na kryterium wartości krytycznej w zaproponowanej przez autorów formie nie wymaga zbioru przycinania, może on być jednak przydatny w celu uniknięcia potencjalnych problemów. W przypadku opracowanego klasyfikatora wykorzystującego klastrowe–rozmyte drzewa decyzyjne oraz klastrowo–kontekstowe rozmyte drzewa decyzyjne zbiór przycinania jest wymagany. Stanowi to odstępstwo od oryginalnych założeń metody, które było konieczne w celu zapewnienia stabilnego i wiarygodnego działania tej metody w opracowanym klasyfikatorze.

Przycinania drzewa w opisany sposób jest szybkie, ale musi ono zostać dokonane dla każdej wartości granicznej, która to jest zmieniana iteracyjnie – liczba iteracji może zaś być znacząca. Sprawia to, że sumaryczny czas działania metody, mimo relatywnie prostych założeń, może się znacząco wydłużyć.

4.2.5 Przycinanie oparte na kryterium kosztu–złożoności

Metoda przycinania opartego na kryterium kosztu–złożoności (ang. *Cost–Complexity Pruning – CCP*) jest znana przede wszystkim z tego, że jest używana w algorytmie CART. Jej koncepcja została zaproponowana w [14].

Koszt błędu węzła p po przycięciu drzewa, oznaczony jako $R(p)$, obliczany jest za pomocą wzoru:

$$R(p) = \frac{g_p}{G_p} \tag{4.7}$$

Koszt błędu nieprzyciętego poddrzewa T_p, oznaczony jako $R(T_p)$, jest obliczany następująco:

$$R(T_p) = \frac{\sum g_l}{G_p} \tag{4.8}$$

Koszt złożoności α (koszt dodatkowego liścia w drzewie [14]) obliczany jest w następujący sposób:

$$\alpha = \frac{R(p) - R(T_p)}{P_{T_p} - 1} \tag{4.9}$$

Algorytm oblicza α dla każdego wewnętrznego węzła drzewa i przycina drzewo w miejscu tego węzła, który posiada największą wartość α. Przycięte drzewo jest zapisywane, a następnie ta sama operacja jest powtarzana dla przyciętego drzewa. Opisany algorytm działa tak długo, aż w drzewie pozostanie jedynie korzeń. Rezultatem działania tej metody jest zbiór drzew, spośród którego, jako ostateczny wynik przycinania, wybierane jest to drzewo, które osiągnęło najwyższą skuteczność klasyfikacji.

Główną wadą opisanej metody jest czas jej działania. Obliczenia współczynnika błędu oraz kosztu złożoności są wymagające czasowo. W każdym kroku algorytmu analizie musi zostać poddane całe drzewo, a koszty obliczane są dla każdego węzła. Co więcej, w każdym drzewie przycinane są jedynie te węzły, które mają najniższy odpowiadający im współczynnik α. Jeśli początkowe drzewo, które miało zostać przycięte przy użyciu tej metody, posiada wiele liści, algorytm przeprowadza dużą liczbę iteracji, a w wyniku jego działania powstaje duży zbiór drzew, z których finalnie tylko jedno zostaje wybrane jako wynik przycinania. Do poprawnego działania sposób ten wymaga zbioru przycinania.

5. Zrównoleglenie obliczeń w ramach rozmytego–klastrowego lasu losowego

W niniejszym rozdziale opisane zostały zagadnienia związane z przeprowadzaniem obliczeń równoległych w ramach opracowanego klasyfikatora. Przedstawiono w nim podstawy teoretyczne dotyczące zagadnienia obliczeń równoległych, omówiono ich istotę oraz motywację do ich przeprowadzania. Opisano także, w jaki sposób zrównoleglone zostały obliczenia wykonywane podczas procesu klasyfikacji danych w rozmytym–klastrowym lesie losowym. Zaproponowano również dalsze prace związane ze zrównoleglaniem obliczeń, które mogą być przeprowadzone w celu udoskonalenia opracowanego rozwiązania.

5.1 Zrównoleglenie obliczeń w rozmytym–klastrowym lesie losowym

Jednym z największych problemów związanych z rozmytym–klastrowym lasem losowym jest jego czas uczenia, w szczególności dla dużych zbiorów danych. Problem ten związany jest z naturą klastrowych–rozmytych drzew decyzyjnych oraz klastrowo–kontekstowych rozmytych drzew decyzyjnych. Drzewa te posiadają tendencję do nadmiernego rozrostu, zwłaszcza dla dużych zbiorów danych, dużej liczby założonych klastrów, na które dzielone są węzły oraz mało restrykcyjnych kryteriów zatrzymania. W rezultacie proces tworzenia tego rodzaju drzew może zająć relatywnie dużo czasu. Problem ten staje się szczególnie zauważalny, gdy klastrowe–rozmyte drzewa decyzyjne oraz klastrowo–kontekstowe rozmyte drzewa decyzyjne mają zostać użyte w klasyfikatorze zbiorczym. Taka sytuacja ma miejsce w przypadku rozmytego–klastrowego lasu losowego.

W swoich założeniach rozmyty–klastrowy las losowy najpierw jest tworzony, a następnie, gdy proces jego tworzenia dobiegnie końca, może zostać wykorzystany do klasyfikacji. W ramach tworzenia klasyfikatora, kolejne klastrowe–rozmyte drzewa decyzyjne lub klastrowo–kontekstowe rozmyte drzewa decyzyjne są tworzone jedno po drugim, niezależnie od siebie. Oznacza to, że proces tworzenia rozmytego–klastrowego lasu losowego, który to proces jest najbardziej czasochłonną częścią jego działania, może zostać w pełni zrównoleglony. Każde drzewo wchodzące w skład tego klasyfikatora zbiorczego może zostać stworzone oddzielnie, niezależnie od pozostałych, a następnie dołączone do lasu. Tworzenie drzew może odbywać się na wielu jednostkach obliczeniowych równocześnie.

Pełne zrównoleglenie procesu uczenia rozmytego–klastrowego lasu losowego zakłada, że w najbardziej optymistycznej sytuacji czas uczenia całego lasu może zostać sprowadzony do czasu uczenia pojedynczego drzewa wchodzącego w jego skład (przy założeniu, że maszyna, na której tworzony jest klasyfikator, posiadałaby tyle jednostek obliczeniowych, ile drzew ma wchodzić w skład lasu, a czas komunikacji między tymi jednostkami byłby pomijalny). Oczywistym jest, że osiągnięcie idealnego przyspieszenia zbliżonego do liniowego jest praktycznie nieprawdopodobne – tym niemniej wydaje się, że zrównoleglenie procesu tworzenia rozmytego–klastrowego lasu losowego znacząco skróci czas jego trwania.

Należy też zauważyć, że w pełnym cyklu działania opracowanej w ramach niniejszej pracy metody da się wyróżnić także inne części, które z powodzeniem mogłyby zostać zrównoleglone. Przykładowo, dobrymi kandydatami do przeprowadzenia obliczeń w sposób równoległy wydają się być procesy przycinania drzew oraz uzyskiwania wag operatorów uporządkowanego ważonego uśredniania. Badania nad zrównolegleniem obliczeń przeprowadzone w niniejszej pracy mają na celu pokazać potencjał stworzonego klasyfikatora do optymalizacji i z całą pewnością nie wyczerpują tematu. Do zaprezentowania potencjału klasyfikatora w tej kwestii wybrany został proces nauki poszczególnych drzew w ramach klasyfikatora, co nie oznacza, że inne części algorytmu także nie mogą zostać zrównoleglone. Wydzielenie takich części może z powodzeniem być tematem dalszych badań nad opracowaną metodą.

6. Podsumowanie i wnioski

W ramach niniejszej rozprawy stanięto przed zadaniami klasyfikacji oraz, dodatkowo, regresji danych. Zaproponowano nowy sposób na zmierzenie się z tymi problemami. Jest nim klasyfikator zbiorczy – rozmyty–klastrowy las losowy wykorzystujący klastrowe–rozmyte drzewa decyzyjne oraz klastrowo–kontekstowe drzewa decyzyjne. Stworzenie klasyfikatora zbiorczego opartego o klastrowe–rozmyte drzewa decyzyjne, który byłby w stanie osiągnąć dobre wyniki dla szerokiego spektrum problemów klasyfikacji, a także, dodatkowo, regresji danych, było celem niniejszej rozprawy.

Badania przeprowadzone i opisane w niniejszej pracy obracały się wokół opracowanej metody klasyfikacji i regresji danych. Na początku klasyfikator został stworzony w dwóch wersjach – wykorzystujących klastrowe–rozmyte drzewa decyzyjne oraz klastrowo–kontekstowe rozmyte drzewa decyzyjne. Pierwsza wersja była w swoich założeniach dedykowana do zadań klasyfikacji danych, druga zaś do regresji – choć obie wersje potrafią poradzić sobie z oboma wspomnianymi problemami.

Opracowana metoda została następnie rozwinięta o możliwość ważonego podejmowania decyzji. Zaproponowano trzy autorskie sposoby wykorzystania operatorów uporządkowanego ważonego uśredniania do wykorzystywania całej wiedzy zawartej w klastrowym–rozmytym lesie losowym z klastrowymi–rozmytymi drzewami decyzyjnymi – lokalne OWA, globalne OWA dla drzewa oraz globalne OWA dla lasu – w celu ważenia ostatecznej decyzji podejmowanej przez klasyfikator zbiorczy. Kolejne dwie autorskie metody – ważenie proporcjonalne na podstawie indeksu strukturalności drzewa oraz na podstawie wag uzyskanych ze zbioru uczącego – opracowano dla wersji klasyfikatora z klastrowo–kontekstowymi rozmytymi drzewami decyzyjnymi. W ten sposób wzbogacono klasyfikator o możliwość wykorzystania zgromadzonej w nim wiedzy, która przepadałaby w procesach klasyfikacji i regresji, gdyby metody te nie zostały opracowane.

Kolejnym krokiem w rozwoju opracowanej metody było dostosowanie metod przycinania drzew decyzyjnych do stworzonego rozwiązania. Umożliwiono wykorzystanie pięciu metod przycinania utworzonych drzew w ramach rozmytego–klastrowego lasu losowego z klastrowymi–rozmytymi drzewami decyzyjnymi oraz trzech w wersji klasyfikatora z klastrowo–kontekstowymi drzewami decyzyjnymi. Pozwoliło to na redukcję rozmiaru drzew wchodzących w skład opracowanego lasu, co umożliwiło skrócenie czasu klasyfikacji oraz regresji, zwiększenie jakości tych procesów oraz poprawę stopnia klarowności wiedzy przechowywanej w drzewach.

Zaproponowano również sposób na przyspieszenie czasu konstrukcji rozmytego–klastrowego lasu losowego poprzez zrównoleglenie obliczeń wykonywanych w procesie jego tworzenia. Wykazano, że utworzony klasyfikator posiada znaczny potencjał do zrównoleglenia, co pozwala na znaczne przyspieszenie tych procesów w ramach jego działania, które są szczególnie czasochłonne.

Przeprowadzono liczne eksperymenty mające na celu ewaluację działania opracowanej metody. Sprawdzono efektywność działania samego klasyfikatora oraz każdego z wprowadzonych usprawnień. Eksperymenty wykazały, że utworzone rozwiązanie daje bardzo dobre wyniki w porównaniu do popularnych i cenionych drzew decyzyjnych. Pokazały one siłę klasyfikatora zbiorczego w zadaniach zarówno klasyfikacji, jak i regresji. Wykazały, że aspekt losowości zaszyty w koncepcję rozwiązania w rzeczywisty sposób zwiększa jakość otrzymanych rezultatów. Pozwoliły one na stwierdzenie, że wykorzystanie wag w procesie podejmowania decyzji utylizuje wiedzę przechowywaną przez klasyfikator, poprawiając osiągnięte wyniki. Umożliwiły także pozytywną ocenę efektów redukcji rozmiaru utworzonych drzew wchodzących w skład lasu przy użyciu metod przycinania. Ujawniły również potencjał klasyfikatora do zrównoleglenia przeprowadzanych w jego ramach obliczeń.

Wszystkie omówione powyżej aspekty pozwalają na pozytywną weryfikację wszystkich postawionych hipotez badawczych. Cel rozprawy został osiągnięty – opracowana metoda sprawdza się bardzo dobrze w zadaniach klasyfikacji, a także regresji.

Przeprowadzone w ramach niniejszej rozprawy badania nie wyczerpują oczywiście tematu. Opracowana metoda może być w dalszym ciągu rozwijana oraz stanowić inspirację do opracowania kolejnych rozwiązań z zakresu klasyfikacji i regresji danych.

Bibliografia

[1] N. Alajlan, Y. Bazi, H. Alhichri, F. Melgani i R. R. Yager. Using OWA Fusion Operators for the Classification of Hyperspectral Images. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, 6(2):602–614, 2013.

[2] S. Bernard, S. Adam i L. Heutte. Dynamic random forests. *Pattern Recognition Letters*, 33(12):1580 – 1586, 2012.

[3] J. Bezdek i S. K. Pal. Fuzzy models for pattern recognition : methods that search for structures in data / edited by J. C. Bezdek, S. K. Pal. *IEEE Press*, 1992.

[4] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 1981.

[5] G. Biau. Analysis of a Random Forests Model. *J. Mach. Learn. Res.*, 13(1):1063–1095, 2012.

[6] L. Biau, G.and Devroye i G. Lugosi. Consistency of Random Forests and Other Averaging Classifiers. *J. Mach. Learn. Res.*, 9:2015–2033, 2008.

[7] L. Bobrowski i J. C. Bezdek. C-means Clustering with the Ll and L? Norms. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(3):545–554, 1991.

[8] P. P. Bonissone, J. M. Cadenas, M. C. Garrido i R. A. Diaz-Valladares. Combination methods in a Fuzzy Random Forest. In *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*, pages 1794–1799, 2008.

[9] P. P. Bonissone, Jose M. Cadenas, M. C. Garrido i R. A. Diaz-Valladares. A fuzzy random forest. *International Journal of Approximate Reasoning*, 51(7):729 – 747, 2010.

[10] P. P. Bonissone, M. C. Garrido J. M. Cadenas i R. A. Diaz-Valladares. A fuzzy random forest: Fundamental for design and construction. In *In Proceedings of the 12th International Conference on Information Processing and Management of Uncertainty in Knowledge- Based Systems (IPMU'08)*, pages 1231–1238, 2008.

[11] L. Breiman. Bagging Predictors. *Mach. Learn.*, 24(2):123–140, 1996.

[12] L. Breiman. Using adaptive bagging to debias regressions. Technical report, Statistics Department, University of California, Berkeley, 1999.

[13] L. Breiman. Random Forests. *Mach. Learn.*, 45(1):5–32, 2001.

[14] L. Breiman, J. Friedman, R. Olshen i C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984.

[15] L. Breiman, J. Friedman, R. Olshen i C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984.

[16] J. M. Cadenas, M. C. Garrido i R. Martinez. Learning in a Fuzzy Random Forest ensemble from imperfect data. In *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, pages 277–282, 2011.

[17] J. M. Cadenas, M. C. Garrido, R. Martinez i P. P. Bonissone. Towards the learning from low quality data in a Fuzzy Random Forest ensemble. In *Fuzzy Systems (FUZZ), 2011 IEEE International Conference on*, pages 2897–2904, 2011.

[18] J. M. Cadenas, M. C. Garrido, R. Martínez i P. P. Bonissone. Extending information processing in a Fuzzy Random Forest ensemble. *Soft Computing*, 16(5):845–861, 2012.

[19] J.M. Cadenas, M.C. Garrido, A. Martinez i R. Martinez. Consensus operators for decision making in Fuzzy Random Forest ensemble. In *Intelligent Systems Design and Applications (ISDA), 2011 11th International Conference on*, pages 1377–1382, 2011.

[20] B. Cestnik i I. Bratko. On estimating probabilities in tree pruning. In *Machine Learning — EWSL-91*, pages 138–150, Berlin, Heidelberg, 1991. Springer Berlin Heidelberg.

[21] R. L. P. Chang i T. Pavlidis. Fuzzy Decision Tree Algorithms. *Systems, Man and Cybernetics, IEEE Transactions on*, 7(1):28–35, 1977.

[22] J. Chen, K. Li, Z. Tang, K. Bilal, S. Yu, C. Weng i K. Li. A Parallel Random Forest Algorithm for Big Data in a Spark Cloud Computing Environment. *IEEE Transactions on Parallel and Distributed Systems*, 28(4):919–933, 2017.

[23] F. Chin-Yuan, C. Pei-Chann, L. Jyun-Jie i J. C. Hsieh. A hybrid model combining case-based reasoning and fuzzy decision tree for medical data classification. *Applied Soft Computing*, 11(1):632 – 644, 2011.

[24] R. N. Dave. Characterization and Detection of Noise in Clustering. *Pattern Recogn. Lett.*, 12(11):657–664, 1991.

[25] R. N. Dave i R. Krishnapuram. M-estimators and robust fuzzy clustering. In *Proceedings of North American Fuzzy Information Processing*, pages 400–404, 1996.

[26] R. N. Dave i R. Krishnapuram. Robust Clustering Methods: A Unified View. *Trans. Fuz Sys.*, 5(2):270–293, 1997.

[27] R. N. Dave i S. Sen. Generalized noise clustering as a robust fuzzy c-M-estimators model. In *1998 Conference of the North American Fuzzy Information Processing Society - NAFIPS (Cat. No.98TH8353)*, pages 256–260, 1998.

[28] M. Denil, D. Matheson i N. De Freitas. Narrowing the Gap: Random Forests in Theory and in Practice. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages I–665–I–673. JMLR.org, 2014.

[29] J. C. Dunn. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3:32–57, 1974.

[30] Sanaa Elyassami i Ali Idri. Applying Fuzzy ID3 Decision Tree for Software Effort Estimation. *CoRR*, abs/1111.0158, 2011.

[31] F. Esposito, D. Malerba, G. Semeraro i J. Kay. A comparative analysis of methods for pruning decision trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):476–491, 1997.

[32] D. Filev i R. R. Yager. Learning OWA operator weights from data. In *Fuzzy Systems, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the Third IEEE Conference on*, pages 468–473 vol.1, 1994.

[33] Y. Freund i R. E. Schapire. A Decision–theoretic Generalization of On-line Learning and an Application to Boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, 1997.

[34] H. Frigui i R. Krishnapuram. A Robust Algorithm for Automatic Extraction of an Unknown Number of Clusters from Noisy Data. *Pattern Recogn. Lett.*, 17(12):1223–1232, 1996.

[35] H. Frigui i R. Krishnapuram. Clustering by Competitive Agglomeration. *Pattern Recogn.*, 30(7):1109–1119, 1997.

[36] H. Frigui i R. Krishnapuram. A robust competitive clustering algorithm with applications in computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):450–465, 1999.

[37] G. Daqi, M. Chen, and Y. Li. A single-layer radial basis function network classifier and its applications. In *Neural Networks, 2005. IJCNN '05. Proceedings. 2005 IEEE International Joint Conference on*, volume 2, pages 1045–1050 vol. 2, 2005.

[38] Ł. Gadomer i Z. A. Sosnowski. Pruning trees in c–fuzzy random forest. *W przygotowaniu*.

[39] Ł. Gadomer i Z. A. Sosnowski. Fuzzy Random Forest with C—Fuzzy Decision Trees. In *Computer Information Systems and Industrial Management 2016, LNCS 9842*, pages 481–492, 2016.

[40] Ł. Gadomer i Z. A. Sosnowski. Using Cluster–Context Fuzzy Decision Trees in Fuzzy Random Forest. In *Computer Information Systems and Industrial Management 2017, LNCS 10244*, pages 180–192, 2017.

[41] Ł. Gadomer i Z. A. Sosnowski. Knowledge aggregation in decision-making process with C-fuzzy random forest using OWA operators. *Soft Computing*, doi:10.1007/s00500-018-3036-x.

[42] Łukasz Gadomer i Zenon A. Sosnowski. Parallel c–fuzzy random forest. In Khalid Saeed i Władysław Homenda, editors, *Computer Information Systems and Industrial Management*, pages 254–265, Cham, 2018. Springer International Publishing.

[43] M. C. Garrido, J. M. Cadenas i P. P. Bonissone. A classification and regression technique to handle heterogeneous and imperfect information. *Soft Computing*, 14(11):1165–1185, 2010.

[44] I. Gath, A. Smolyak Iskoz i B. Van Cutsem. Data induced metric and fuzzy clustering of non-convex patterns of arbitrary shape. *Pattern Recognition Letters*, 18(6):541 – 553, 1997.

[45] R. Genuer. Variance reduction in purely random forests. *Journal of Nonparametric Statistics*, 2:18 – 562, 2012.

[46] P. J. F. Groenen, U. Kaymak i J. van Rosmalen. *Fuzzy Clustering with Minkowski Distance Functions*, chapter 3, pages 53–68. Wiley-Blackwell, 2007.

[47] Patrick J. F. Groenen i Krzysztof Jajuga. Fuzzy clustering with squared Minkowski distances. *Fuzzy Sets and Systems*, 120:227–237, 2001.

[48] D. E. Gustafson i W. C. Kessel. Fuzzy clustering with a fuzzy covariance matrix. In *1978 IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes*, pages 761–766, 1978.

[49] J. Han, M. Kamber i J. Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2011.

[50] F. Hopner, F. Hoppner i F. Klawonn. *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition*. Wiley & Sons, 1999.

[51] C. Z. Janikow. Fuzzy decision trees: issues and methods. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 28(1):1–14, 1998.

[52] K. Jurczuk, M. Czajkowski i M. Kretowski. Evolutionary induction of a decision tree for large-scale data: a gpu-based approach. *Soft Computing*, 21(24):7363–7379, 2017.

[53] K. Jurczuk, M. Kretowski i J. Bezy-Wendling. GPU-based computational modeling of magnetic resonance imaging of vascular structures. *The International Journal of High Performance Computing Applications*, 32(4):496–511, 2018.

[54] A. Keller. Fuzzy clustering with outliers. In *PeachFuzz 2000. 19th International Conference of the North American Fuzzy Information Processing Society - NAFIPS (Cat. No.00TH8500)*, pages 143–147, 2000.

[55] F. Klawonn i F. Hoppner. What is fuzzy about fuzzy clustering? understanding and improving the concept of the fuzzifier. In Michael R. Berthold, Hans-Joachim

Lenz, Elizabeth Bradley, Rudolf Kruse i Christian Borgelt, editors, *Advances in Intelligent Data Analysis V*, pages 254–264, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[56] F. Klawonn, R. Kruse i H. Timm. Fuzzy shell cluster analysis. In Giacomo Della Riccia, Hans-Joachim Lenz i Rudolf Kruse, editors, *Learning, Networks and Statistics*, pages 105–119, Vienna, 1997. Springer Vienna.

[57] R. Krishnapuram i J. M. Keller. The possibilistic C-means algorithm: insights and recommendations. *IEEE Transactions on Fuzzy Systems*, 4(3):385–393, 1996.

[58] R. Kruse, Ch. Doring i M.-J. Lesot. *Fundamentals of Fuzzy Clustering*, chapter 1, pages 1–30. Wiley-Blackwell, 2007.

[59] A. Kumar, M. Hanmandlu i H. M. Gupta. Fuzzy binary decision tree for biometric based personal authentication. *Neurocomputing*, 99:87 – 97, 2013.

[60] V. Kumar. *Introduction to Parallel Computing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 2002.

[61] L. C. Jain, and G. N. Allen. Introduction to artificial neural networks. In *Electronic Technology Directions to the Year 2000, 1995. Proceedings.*, pages 36–62, 1995.

[62] Y. Lertworaprachaya, Y. Yang i R. John. Interval–valued fuzzy decision trees with optimal neighbourhood perimeter. *Applied Soft Computing*, 24:851 – 866, 2014.

[63] M. Lichman. UCI machine learning repository, 2013.

[64] W. Limprasert. Parallel random forest with IPython cluster. In *2015 International Computer Science and Engineering Conference (ICSEC)*, pages 1–6, 2015.

[65] X. Liu, X. Feng i W. Pedrycz. Extraction of fuzzy rules from fuzzy decision trees: An axiomatic fuzzy sets (afs) approach. *Data and Knowledge Engineering*, 84:1 – 25, 2013.

[66] X. Liu i W. Pedrycz. The development of fuzzy decision trees in the framework of axiomatic fuzzy set logic. *Applied Soft Computing*, 7(1):325 – 342, 2007.

[67] Z. G. Liu, Q. Pan, J. Dezert i A. Martin. Combination of Classifiers With Optimal Weight Based on Evidential Reasoning. *IEEE Transactions on Fuzzy Systems*, 26(3):1217–1230, 2018.

[68] J. Macqueen. Some methods for classification and analysis of multivariate observations. In *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[69] J. M. Merigo, A. M. Gil-Lafuente, D. Yu i C.s Llopis-Albert. Fuzzy decision making in complex frameworks with generalized aggregation operators. *Applied Soft Computing*, 68:314 – 321, 2018.

[70] R. Mesiar, A. Stupnanova i R. R. Yager. Generalizations of OWA operators. *Fuzzy Systems, IEEE Transactions on*, 23(6):2154–2162, 2015.

[71] J. Mingers. Expert systems—rule induction with statistical data. *Journal of the Operational Research Society*, 38(1):39–47, 1987.

[72] M. Mizumoto. Fuzzy controls under various fuzzy reasoning methods. *Information Sciences*, 45(2):129 – 151, 1988.

[73] F. Mota i F. Gomide. Fuzzy Clustering in Fitness Estimation Models for Genetic Algorithms and Applications. In *2006 IEEE International Conference on Fuzzy Systems*, pages 1388–1395, 2006.

[74] N. Landwehr, M. Hall, and E. Frank. Logistic Model Trees. *Mach. Learn.*, 59(1-2):161–205, 2005.

[75] Sa. Nefti-Meziani i M. Oussalah. *Inclusion-based Fuzzy Clustering*, chapter 9, pages 193–209. Wiley-Blackwell, 2007.

[76] T. Niblett i I Bratko. Learning Decision Rules in Noisy Domains. In *Proceedings of Expert Systems '86, The 6Th Annual Technical Conference on Research and Development in Expert Systems III*, pages 25–34, New York, NY, USA, 1987. Cambridge University Press.

[77] N. R. Pal, K. Pal i J. C. Bezdek. A mixed c-means clustering model. In *Proceedings of 6th International Fuzzy Systems Conference*, volume 1, pages 11–21, 1997.

[78] N. R. Pal, K. Pal, J. M. Keller i J. C. Bezdek. A new hybrid c-means clustering model. In *2004 IEEE International Conference on Fuzzy Systems (IEEE Cat. No.04CH37542)*, volume 1, pages 179–184, 2004.

[79] D. A. Patterson i J. L. Hennessy. *Computer Organization and Design, Fifth Edition: The Hardware/Software Interface*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 5th edition, 2013.

[80] W. Pedrycz. Conditional Fuzzy C-Means. *Pattern Recogn. Lett.*, 17(6):625–631, 1996.

[81] W. Pedrycz i Z. A. Sosnowski. C–fuzzy decision trees. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 35(4):498–511, 2005.

[82] J. R. Quinlan. Induction of Decision Trees. *Mach. Learn.*, 1(1):81–106, 1986.

[83] J. R. Quinlan. Simplifying Decision Trees. *Int. J. Man-Mach. Stud.*, 27(3):221–234, 1987.

[84] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[85] M. V. Ribeiro, H. A. Camargo i M. E. Cintra. A comparative analysis of pruning strategies for fuzzy decision trees. In *2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS)*, pages 709–714, 2013.

[86] J. J. Rodriguez, L. I. Kuncheva i C. J. Alonso. Rotation Forest: A New Classifier Ensemble Method. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(10):1619–1630, 2006.

[87] P. J. Rousseeuw, E. Trauwaert i L. Kaufman. Fuzzy clustering with high contrast. *Journal of Computational and Applied Mathematics*, 64(1):81 – 90, 1995.

[88] D. Rutkowska, M. Piliński i L. Rutkowski. *Sieci neuronowe, algorytmy genetyczne i systemy rozmyte*. Wydaw. Naukowe PWN, 1997.

[89] H. Sahbi i N. Boujemaa. Validity of Fuzzy Clustering Using Entropy Regularization. In *The 14th IEEE International Conference on Fuzzy Systems, 2005. FUZZ '05.*, pages 177–182, 2005.

[90] S. Sardari, M. Eftekhari i F. Afsari. Hesitant fuzzy decision tree approach for highly imbalanced data classification. *Applied Soft Computing*, 61:727 – 741, 2017.

[91] L. R. Scariot da Silva, F. Gomide i R. Yager. *Fuzzy Clustering with Participatory Learning and Applications*, chapter 7, pages 137–153. Wiley-Blackwell, 2007.

[92] B. Scholkopf i A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, Cambridge, MA, USA, 2001.

[93] S. K. Shukla i M. K. Tiwari. GA Guided Cluster Based Fuzzy Decision Tree for Reactive Ion Etching Modeling: A Data Mining Approach. *IEEE Transactions on Semiconductor Manufacturing*, 25(1):45–56, 2012.

[94] Z. A. Sosnowski. Reguły decyzyjne z rozmytą granulacją wiedzy. *Symulacja w Badaniach i Rozwoju*, Vol. 3, nr 4:225–232, 2012.

[95] H. Timm, C. Borgelt, Ch. Daring i R. Kruse. An extension to possibilistic fuzzy cluster analysis. *Fuzzy Sets and Systems*, 147(1):3 – 16, 2004.

[96] H. Timm i R. Kruse. A modification to improve possibilistic fuzzy cluster analysis. In *2002 IEEE World Congress on Computational Intelligence. 2002 IEEE International Conference on Fuzzy Systems. FUZZ-IEEE'02. Proceedings (Cat. No.02CH37291)*, volume 2, pages 1460–1465, 2002.

[97] M. Umanol, H. Okamoto, I. Hatono, H. Tamura, F. Kawachi, S. Umedzu i J. Kinoshita. Fuzzy decision trees by fuzzy ID3 algorithm and its application to diagnosis systems. In *Proceedings of 1994 IEEE 3rd International Fuzzy Systems Conference*, volume 3, pages 2113–2118, 1994.

[98] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[99] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, Berlin, Heidelberg, 1995.

[100] X. Wang, Xi. Liu, W. Pedrycz i L. Zhang. Fuzzy rule based decision trees. *Pattern Recognition*, 48(1):50 – 59, 2015.

[101] X. Z. Wang, L. C. Dong i J. H. Yan. Maximum Ambiguity-Based Sample Selection in Fuzzy Decision Tree Induction. *IEEE Transactions on Knowledge and Data Engineering*, 24(8):1491–1505, 2012.

[102] Y. Wang, S. Xia, Q. Tang, J. Wu i X. Zhu. A Novel Consistent Random Forest Framework: Bernoulli Random Forests. *IEEE Transactions on Neural Networks and Learning Systems*, 29(8):3510–3523, 2018.

[103] R. Weber. Fuzzy-ID3: A class of methods for automatic knowledge acquisition. In *2. Int. Conf. on Fuzzy Logic Neural Networks. Iizuka, Japan 1992*, pages 265–268, 1992.

[104] I. H. Witten, E. Frank i M. A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2011.

[105] K. Wu i M. Yang. Alternative c-means clustering algorithms. *Pattern Recognition*, 35:2267–2278, 2002.

[106] Z. Wu, W. Xie i J. Yu. Fuzzy C-means clustering algorithm based on kernel method. In *Proceedings Fifth International Conference on Computational Intelligence and Multimedia Applications. ICCIMA 2003*, pages 49–54, 2003.

[107] J. Xia, S. Zhang, G. Cai, L. Li, Q. Pan, J. Yan i G. Ning. Adjusted weight voting algorithm for random forests in handling missing values. *Pattern Recognition*, 69:52 – 60, 2017.

[108] R. R. Yager. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *Systems, Man and Cybernetics, IEEE Transactions on*, 18(1):183–190, 1988.

[109] R. R. Yager. A model of participatory learning. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(5):1229–1234, 1990.

[110] R. R. Yager i N. Alajlan. Probabilistically Weighted OWA Aggregation. *Fuzzy Systems, IEEE Transactions on*, 22(1):46–56, 2014.

[111] R. R. Yager i G. Beliakov. OWA Operators in Regression Problems. *Fuzzy Systems, IEEE Transactions on*, 18(1):106–113, 2010.

[112] R. R. Yager, M. Z. Reformat i G. Gumrah. Fuzziness, OWA and linguistic quantifiers for web selection processes. In *Fuzzy Systems (FUZZ), 2011 IEEE International Conference on*, pages 1751–1758, 2011.

[113] Y. Yuan i M. J. Shaw. Induction of Fuzzy Decision Trees. *Fuzzy Sets Syst.*, 69(2):125–139, 1995.

[114] D. Zhang i S. Chen. Clustering Incomplete Data Using Kernel-Based Fuzzy C-means Algorithm. *Neural Process. Lett.*, 18(3):155–162, 2003.

[115] D. Zhang i S. Chen. Kernel-based fuzzy and possibilistic c-means clustering. In *In International Conference on Artificial Neural Networks (ICANN03*, pages 122–125, 2003.

[116] L. Zhang i P. N. Suganthan. Random Forests with ensemble of feature spaces. *Pattern Recognition*, 47(10):3429–3437, 2014.