

Autoreferat

1 Imię i nazwisko

Jan Kozak

2 Posiadane dyplomy i stopnie naukowe

- *2002 – 2005, Uniwersytet Śląski, Wydział Informatyki i Nauki o Materiałach*
Studia licencjackie, tryb dzienny,
kierunek: Informatyka,
specjalizacja: Systemy informacyjne i systemy inteligentne.
Tematem pracy licencjackiej był system zarządzania treścią strony.
- *2005 – 2007, Uniwersytet Śląski, Wydział Informatyki i Nauki o Materiałach*
Studia magisterskie uzupełniające, tryb dzienny,
kierunek: Informatyka,
specjalizacja: Bazy danych i systemy inteligentne.
Tematem pracy magisterskiej było zagadnienie znajdowania wzorców i reguł, a jej część praktyczna obejmowała nowatorski algorytm genetyczny zaimplementowany przy użyciu języka Python.
Studia ukończone z wynikiem bardzo dobrym.
- *2007 – 2011, Uniwersytet Śląski, Wydział Informatyki i Nauki o Materiałach*
Studia doktoranckie, tryb dzienny,
dyscyplina naukowa: Informatyka.
- *2011, Uzyskanie tytułu doktora nauk technicznych w zakresie informatyki, obrona z wyróżnieniem.*
Uniwersytet Śląski, Wydział Informatyki i Nauki o Materiałach. Rozprawa doktorska napisana pod kierunkiem dr hab. Urszuli Boryczki dotyczyła przede wszystkim analizy algorytmu znanego pod nazwą Ant-Miner oraz wstępnego opracowania algorytmu do konstruowania drzew decyzyjnych.

3 Informacje o dotychczasowym zatrudnieniu w jednostkach naukowych

- *02.2013 – teraz, Uniwersytet Ekonomiczny, Katowice*
Adiunkt w Katedrze Inżynierii Wiedzy; Wydział Informatyki i Komunikacji.
- *11.2016 – 12.2016, King Abdullah University of Science and Technology, Arabia Saudyjska*
Pobyt na stażu w Electrical and Mathematical Science and Engineering Division.
- *06.2011 – 09.2016, Uniwersytet Śląski, Katowice*
Adiunkt w Zakładzie Algorytmiki i Inteligencji Obliczeniowej; Instytut Informatyki; Wydział Informatyki i Nauki o Materiałach.
- *10.2007 – 06.2011, Uniwersytet Śląski, Katowice*
Asystent naukowo-dydaktyczny w Zakładzie Systemów Informatycznych; Instytut Informatyki; Wydział Informatyki i Nauki o Materiałach.

4 Podstawowe osiągnięcie naukowo-badawcze

Jako podstawowe osiągnięcie naukowe przedstawiam cykl 12 publikacji monotematycznych, na które składa się monografia [M] będąca uwieńczeniem pozostałych publikacji w postaci 3 artykułów z listy JCR

([A1] — [A3]) o sumarycznym Impact Factor 11,014, a także 8 artykułów z innych czasopism oraz konferencji, z których recenzowane artykuły w większości indeksowane są w bazach WoS lub Scopus ([B1] — [B8]). Oświadczenia związane z opisem wkładu pracy współautorów oraz jego oszacowaniu procentowym zostały załączone do wniosku.

[M] J. Kozak. *Decision Tree and Ensemble Learning Based on Ant Colony Optimization*. Studies in Computational Intelligence, tom 781, edytor serii prof. Janusz Kacprzyk, 159 stron, Springer International Publishing AG 2018, ISSN 978-3-319-93751-9, ISBN 1860-949X, 2018.
doi.org/10.1007/978-3-319-93752-6
[J. Kozak 100%] (MNiSW 25 punktów)

Publikacje będące częścią monografii:

- [A1] J. Kozak, U. Boryczka. *Collective data mining in the ant colony decision tree approach*. Information Sciences, tom 372, strony 126–147, Elsevier, 2016.
doi.org/10.1016/j.ins.2016.08.051
[J. Kozak 80%, U. Boryczka 20%] (IF=4,832; MNiSW 45 punktów)
- [A2] J. Kozak, U. Boryczka. *Multiple Boosting in the Ant Colony Decision Forest Meta-Classifer*. Knowledge-Based Systems, tom 75, strony 141–151, Elsevier, 2015.
doi:10.1016/j.knosys.2014.11.027
[J. Kozak 75%, U. Boryczka 25%] (IF=3,325; MNiSW 35 punktów)
- [A3] U. Boryczka, J. Kozak. *Enhancing the effectiveness of Ant Colony Decision Tree algorithms by co-learning*. Applied Soft Computing Journal, tom 30, strony 166–178, Elsevier, 2015.
doi:10.1016/j.asoc.2014.12.036
[J. Kozak 60%, U. Boryczka 40%] (IF=2,857; MNiSW 40 punktów)
- [B1] J. Kozak, P. Juszczuk. *Association ACDT as a tool for discovering the financial data rules*. INISTA, IEEE International Conference on INnovations in Intelligent SysTems and Applications, strony 241–246, 2017.
doi.org/10.1109/INISTA.2017.8001164
[J. Kozak 70%, P. Juszczuk 30%] (MNiSW 15 punktów)
- [B2] U. Boryczka, B. Probierz, J. Kozak. *Adaptive Ant Colony Decision Forest in Automatic Categorization of Emails*. Intelligent Information and Database Systems – 7th Asian Conference, ACIIDS, tom 9011 serii Lecture Notes in Computer Science, strony 451–461. Springer, 2015.
doi.org/10.1007/978-3-319-15702-3_44
[J. Kozak 50%, U. Boryczka 20%, B. Probierz 30%] (MNiSW 15 punktów)
- [B3] J. Kozak, U. Boryczka. *Goal-oriented Requirements for ACDT Algorithms*. Computational Collective Intelligence. Technologies and Applications, ICCCI, tom 8733 serii Lecture Notes in Computer Science, strony 593–602. Springer, 2014.
doi.org/10.1007/978-3-319-11289-3_60
[J. Kozak 80%, U. Boryczka 20%] (MNiSW 10 punktów)
- [B4] U. Boryczka, J. Kozak. *On-the-go adaptability in the new Ant Colony Decision Forest Approach*. ACIIDS (2), tom 8398 serii Lecture Notes in Computer Science, strony 157–166, Springer, Berlin, 2014.
doi.org/10.1007/978-3-319-05458-2_17
[J. Kozak 80%, U. Boryczka 20%] (MNiSW 10 punktów)
- [B5] J. Kozak, U. Boryczka. *Dynamic version of the ACDT/ACDF algorithm for H-bond data set analysis*. Computational Collective Intelligence. Technologies and Applications, ICCCI, tom 8083 serii Lecture Notes in Computer Science, strony 701–710. Springer, 2013.
doi.org/10.1007/978-3-642-40495-5_70
[J. Kozak 90%, U. Boryczka 10%] (MNiSW 10 punktów)
- [B6] U. Boryczka, J. Kozak. *Ant Colony Decision Forest Meta-ensemble*. Computational Collective Intelligence. Technologies and Applications, ICCCI (2), tom 7654 serii Lecture Notes in Computer Science, strony 473–482. Springer, Berlin, 2012.
doi.org/10.1007/978-3-642-34707-8_48
[J. Kozak 80%, U. Boryczka 20%] (MNiSW 10 punktów)

- [B7] U. Boryczka, J. Kozak. *New insights of cooperation among ants in Ant Colony Decision Trees*. Third World Congress on Nature and Biologically Inspired Computing, NaBIC, strony 262–267, IEEE, 2011.
<https://doi.org/10.1109/NaBIC.2011.6089463>
 [J. Kozak 80%, U. Boryczka 20%]
- [B8] U. Boryczka, J. Kozak. *An adaptive discretization in the ACDT algorithm for continuous attributes*. Computational Collective Intelligence. Technologies and Applications, ICCCI (2), tom 6923 serii Lecture Notes in Computer Science, strony 475–484. Springer Berlin / Heidelberg, 2011.
doi.org/10.1007/978-3-642-23938-0_48
 [J. Kozak 90%, U. Boryczka 10%] (MNiSW 13 punktów)

4.1 Tytuł osiągnięcia naukowego

Uczenie drzew decyzyjnych oraz zespołów klasyfikatorów z zastosowaniem algorytmów mrowiskowych.

4.2 Wprowadzenie

Jednym z problemów uczenia maszynowego jest problem klasyfikacyjny. Natomiast jedną z metod stosowaną w tej problematyce jest model predykcji nazywany drzewami decyzyjnymi (*ang. decision tree*). Należy podkreślić, że jest to popularny klasyfikator, który może być reprezentowany jako graf skierowany. To ważne, szczególnie z punktu widzenia doboru algorytmów wspomagających rozwiązanie tego problemu. Zasadniczą trudnością w konstruowaniu drzewa decyzyjnego jest wybór podziału danych w każdym węźle drzewa, a więc podjęcie decyzji na każdym etapie jego budowy. Konstruowanie minimalnego pod względem wielkości, optymalnego (w przytaczanym przypadku pod względem liczby testów niezbędnych do zidentyfikowania obiektu) drzewa decyzyjnego jest problemem *NP*-zupełnym [12, 16].

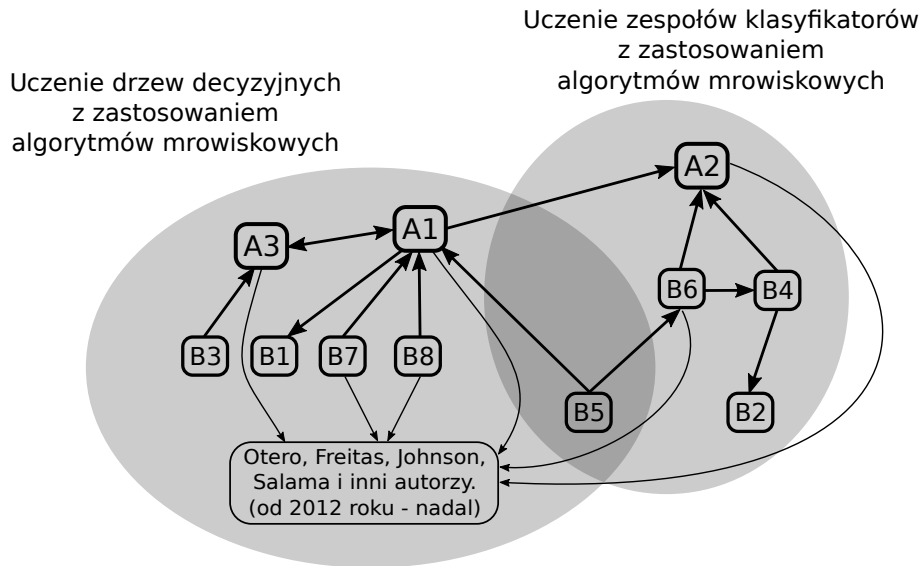
Klasyczne podejście do konstruowania drzew decyzyjnych odbywa się z zastosowaniem metod statystycznych. Jednak właściwość związana z reprezentacją drzew decyzyjnych oraz duża przestrzeń dopuszczalnych rozwiązań wskazują, że dobre rezultaty mogą być uzyskiwane przy zastosowaniu metod przybliżonych. W zakresie inteligencji obliczeniowej jedną z takich metod jest optymalizacja kolonii mrówek (*ang. ant colony optimization, ACO*) [7], będąca rozwinięciem systemu mrowiskowego [6, 8, 9]. Kolonie składają się z agentów, a pojedynczy agent jest postrzegany jako algorytm i dane niezbędne do jego realizacji. Informacje pozyskiwane są ze środowiska i wpływają na zachowanie agenta w trakcie realizacji określonego celu. Na decyzję agenta mają wpływ dwie wartości: wartość użyteczności, będąca śladem feromonowym (pamięć wspólna agentów) oraz heurystyka, uzależniona od rozwiązywanego problemu. Metoda ACO cechuje się uzyskiwaniem dobrych rezultatów przy zastosowaniu do problemów możliwych do zaprezentowania w postaci grafowej.

Te obserwacje są zasadniczą inspiracją tematyki podjętej w tym osiągnięciu naukowym. Zastosowanie drzew decyzyjnych pozwala na stosunkowo proste tworzenie zespołów klasyfikatorów bazujących na drzewach decyzyjnych (tzw. lasów decyzyjnych), czyli zbiorów pewnych drzew decyzyjnych [15]. Algorytmy ACO podczas pojedynczego wykonania generują pewien zestaw potencjalnych rozwiązań, które w wyniku zastosowanych mechanizmów są eliminowane bądź zastępowane. Stworzenie zespołu klasyfikatorów z odpowiednio wyselekcjonowanych rozwiązań pozwala natomiast na doprowadzenie do konsensusu pomiędzy nimi – specyficznego zespołu klasyfikatorów bazującego na optymalizacji mrowiskowej.

Wcześniejsze prace związane z zastosowaniem algorytmów mrowiskowych w klasyfikacji dotyczyły przede wszystkim budowy reguł decyzyjnych, a dokładniej algorytmu nazwanego Ant-Miner [18]. Algorytm ten był przedmiotem moich badań związanych z rozprawą doktorską. Zebrane obserwacje oraz uzyskane wyniki pozwoliły na przedstawienie wniosków związanych z możliwością budowy drzew decyzyjnych z zastosowaniem algorytmów mrowiskowych. Jednak dopiero po zakończeniu prac nad rozprawą doktorską opracowane zostały konkretne rozwiązania związane z tą problematyką i zaproponowany został algorytm do konstruowania drzew decyzyjnych (*ang. ant colony decision tree, ACDT*). W kolejnych etapach pracy, algorytm ACDT był rozwijany, aż wreszcie możliwe było opracowanie algorytmów do budowy zespołów klasyfikatorów opartych na ACO.

Propozycje opisane w niniejszym autoreferacie zyskały uznanie i były modyfikowane oraz rozwijane przez innych naukowców. Pierwszym z algorytmów, którego powstanie było następstwem analizy zaproponowanego przeze mnie algorytmu ACDT był Ant-Tree-Miner opisany przez Otero i innych w pracy [17]. Różnice pomiędzy tymi algorytmami oraz pozostałe publikacje związane z rozpowszechnianiem algorytmu ACDT opisane są w części 4.4 niniejszego autoreferatu.

Na rys. 1 przedstawiony jest mój wkład w problematykę związaną z osiągnięciem naukowym. Oznaczenia przyjęte na tym rysunku odpowiadają publikacjom monotematycznym zgłoszonym do oceny osiągnięcia naukowego. Dodatkowo rysunek wzbogacony jest o publikacje innych autorów związane z tą tematyką. Na wykresie prace powiązane są ze sobą ze względu na wpływ jednych badań na drugie — niekoniecznie rok publikacji jest zgodny z datą rozpoczęcia analizy danego problemu. Dokładny zakres prac podjętych w tych artykułach przedstawiony jest w dalszej części niniejszego autoreferatu. Dzięki przyjętej formie możliwe jest wskazanie powiązań pomiędzy artykułami oraz ich zakresem – związanym z budową drzew decyzyjnych lub zespołów klasyfikatorów.



Rysunek 1: Schemat przedstawiający wkład pracy

4.3 Uczenie drzew decyzyjnych z zastosowaniem algorytmu mrowiskowego

Pierwszy z algorytmów mrowiskowych stosowanych w procesie uczenia drzew decyzyjnych był algorytm ACDT. W wyniku jego pracy powstaje klasyfikator – drzewo decyzyjne. Jest to algorytm niedeterministyczny, a więc każde jego wykonanie przeważnie skutkuje zbudowaniem innego drzewa decyzyjnego. Idea algorytmu ACDT opiera się na zastosowaniu śladu feromonowego reprezentowanego w formie kraędziowej oraz heurystyki zaczerpniętej z klasycznych algorytmów do konstruowania drzew decyzyjnych (pierwotnie z algorytmu CART [2]).

Podjęcie ACDT można przedstawić, jako piątkę:

$$ACDT = \langle (X, A \cup \{c\}), T(S), ants, p_{m, m_L(i,j)}(t), S \rangle, \quad (1)$$

na którą składa się:

$(X, A \cup \{c\})$ — tabela decyzyjna reprezentująca problem, przedstawiana jako dwójka, gdzie X jest zbiorem obiektów, a A jest zbiorem atrybutów, w tym atrybut decyzyjny c ,

$T(S)$ — drzewo decyzyjne będące rozwiązaniem problemu,

oraz elementy systemu mrowiskowego:

$ants$ — liczba mrówek w iteracji,

$p_{m, m_L(i,j)}(t)$ — reguła wyboru rozwiązania w każdym węźle drzewa decyzyjnego w czasie t (wzór (5)),

S — zbiór dopuszczalnych obiektów w węźle.

Ocena jakości drzewa decyzyjnego (rozwiązania) jest obliczana na podstawie:

$$Q(T) = \phi \cdot s(T) + \psi \cdot ev(T, S), \quad (2)$$

gdzie:

$s(T)$ — ocena wielkości drzewa decyzyjnego T ,

$ev(T, S)$ — ocena jakości klasyfikacji elementów ze zbioru S przy pomocy klasyfikatora T ,

ϕ i ψ — parametry odpowiadające za wagę $s(T)$ i $ev(T, S)$.

Ocena wielkości drzewa T w klasycznej wersji określona jest wzorem:

$$s(T) = \frac{1}{n}, \quad (3)$$

gdzie n jest liczbą wszystkich węzłów w drzewie T .

Jako ocenę jakości klasyfikacji w algorytmie ACDT można stosować różne miary, zostało to opracowane i opublikowane w [B3]. Klasycznie jednak przyjęta jest dokładność klasyfikacji:

$$ev(T, S) = \frac{\sum_{c=1}^C (TP_c)}{|S|}, \quad (4)$$

gdzie:

- TP_c – liczba poprawnie sklasyfikowanych obiektów należących do klasy c ,
- $|S|$ – liczba wszystkich obiektów w zbiorze S ,
- C – liczba klas decyzyjnych.

W przypadku algorytmu ACDT podział w każdym węźle odbywa się na podstawie prawdopodobieństwa wyboru (5), czyli iloczynu wartości funkcji heurystycznej ($\eta_{i,j}$) i śladu feromonowego ($\tau_{m,m_L(i,j)}$). Podczas pracy algorytmu wartość śladu feromonowego wpływa na wybór podziału w każdym węźle (poprzez funkcję (5)), co jest związane z niedeterminizmem algorytmu uczącego. Początkowe drzewo decyzyjne zbliżone do powstałego podczas zastosowania samego kryterium podziału (dostosowanego do funkcji heurystycznej) podlega modyfikacjom (początkowo losowym – (5)). W kolejnych iteracjach wprowadzone zmiany (zapamiętywane w postaci śladu feromonowego) zwiększają prawdopodobieństwo na odwzorowanie ulepszonych (pod względem funkcji oceny jakości (2)) drzewa decyzyjnego. W kolejnych iteracjach uzyskane drzewa decyzyjne ulegają dalszym modyfikacjom.

Prawdopodobieństwo wyboru podziału w każdym węźle obliczane jest na podstawie wzoru:

$$p_{m,m_L(i,j)}(t) = \frac{\tau_{m,m_L(i,j)}(t) \cdot \eta_{i,j}^\beta}{\sum_i^a \sum_j^{b_i} \tau_{m,m_L(i,j)}(t) \cdot \eta_{i,j}^\beta}, \quad (5)$$

gdzie:

- m – węzeł nadrzędny,
- $\eta_{i,j}$ – funkcja heurystyczna dla testu atrybutu i i wartości j (wzór (6)),
- $\tau_{m,m_L(i,j)}$ – ślad feromonowy w czasie t dla krawędzi prowadzącej od węzła m do węzła m_L (dla testu atrybutu i o wartości j),
- β – parametr wpływający na względne znaczenie funkcji heurystycznej.

W przypadku algorytmu ACDT wartość funkcji heurystycznej dla każdej pary atrybut-wartość w węźle m wyznaczana jest na podstawie kryterium podziału na dwie części (*ang. twoing rule*) znanej z algorytmu CART [2] dostosowanego do zastosowania w algorytmach mrowiskowych:

$$\forall_{i \in A, j \in V_i} \quad \eta_{i,j} = \frac{P_l P_r}{4} \left[\sum_{c=1}^C |p(c|m_l) - p(c|m_r)| \right]^2, \quad (6)$$

gdzie:

- V_i – zbiór wartości atrybutu i ,
- $p(c|m_l)$ – prawdopodobieństwo wystąpienia klasy decyzyjnej c w węźle m_l (lewym poddrzewie),
- $p(c|m_r)$ – prawdopodobieństwo wystąpienia klasy decyzyjnej c w węźle m_r (prawym poddrzewie),
- P_l / P_r – prawdopodobieństwo przejścia obiektu do węzła m_l lub m_r .

W klasycznej wersji algorytmu ACDT podział danych w każdym węźle odbywa się na podstawie testu równościowego albo przynależnościowego (dla atrybutów dyskretnych). Algorytm ACDT, dostosowany do pracy z atrybutami o wartościach ciągłych, zawiera test nierównościowy:

$$T(x) = \begin{cases} 1, & \text{if } a_j(x) \leq cut \\ 0, & \text{otherwise } (a_j(x) > cut) \end{cases}, \quad (7)$$

gdzie:

- cut – wartość progowa (punkt cięcia),
- a_j – j -a zmienna.

Wprowadzone rozwiązanie wpływa na precyzyjny podział danych w przypadku wartości ciągłych. Pozwala ono na dokonanie podziału danych na podstawie wartości przedziałów występujących w analizowanym zbiorze danych.

Należy pamiętać, że w algorytmie ACDT wybór reguły podziału w odpowiednim węźle jest wykonywany na podstawie pseudolosowej reguły proporcjonalności (znanej z algorytmów mrowiskowych (5)) w zależności od współczynnika eksploracji i eksploatacji q_0 . Pseudolosowa reguła proporcjonalności jest wykonywana tylko wtedy, gdy $q > q_0$ (dla losowej wartości q). W przeciwnym razie wybierany jest podział o maksymalnej wartości iloczynu śladu feromonowego i funkcji heurystycznej:

$$r = \begin{cases} \arg \max_{m_L \in R_m^c} \{[\tau_{m,m_L}(t)] \cdot [\eta_{i,j}]^\beta\} & \text{if } q \leq q_0 \\ J & \text{if } q > q_0 \end{cases}, \quad (8)$$

gdzie J jest losowo wybierany na podstawie prawdopodobieństwa $p_{m,m_L(i,j)}(t)$ (wzór (5)).

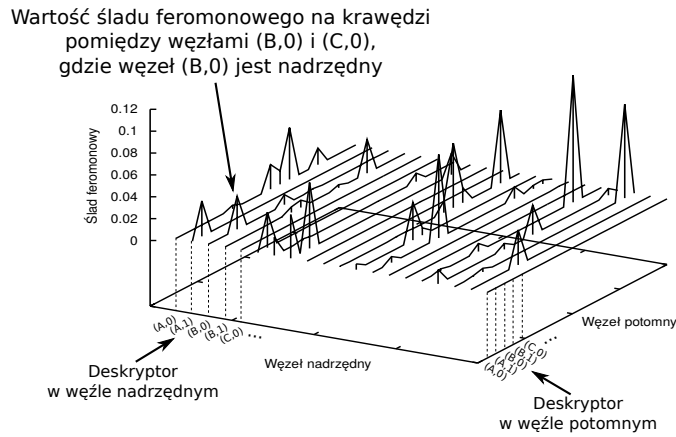
Kolejnym problemem wartym dalszego rozważenia jest modyfikacja wartości śladu feromonowego. Początkowa wartość feromonu ustalana jest na podstawie liczby wartości atrybutów:

$$\tau_{m,m_L}(t=0) = \frac{\log_2(C)}{\sum_{att=1}^{|A|} |a_{att}|}, \quad (9)$$

gdzie:

- $|A|$ – liczba wartości atrybutów,
- $|a_{att}|$ – liczba możliwych wartości atrybutu a_{att} .

Aktualizacja śladu feromonowego polega na zwiększeniu wartości na każdej parze węzłów, które występują w drzewie decyzyjnym (10). Wszystkie możliwe kombinacje krawędzi pomiędzy węzłami przechowywane są w macierzy śladu feromonowego, co zostało zobrazowane na rys. 2. Wyższa wartość śladu feromonowego oznacza silniejsze wzmocnienie dla wybrania takiej krawędzi w konstruowanym drzewie decyzyjnym. W tym modelu zapisywane są tylko krawędzie bez informacji o ich poziomie w drzewie decyzyjnym. Dzięki temu istnieje większa możliwość modyfikacji początkowych struktur drzewa, ponieważ dla danego deskryptora (pary atrybut – wartość) wybór węzła potomnego nie jest ograniczony przez poziom tego drzewa.



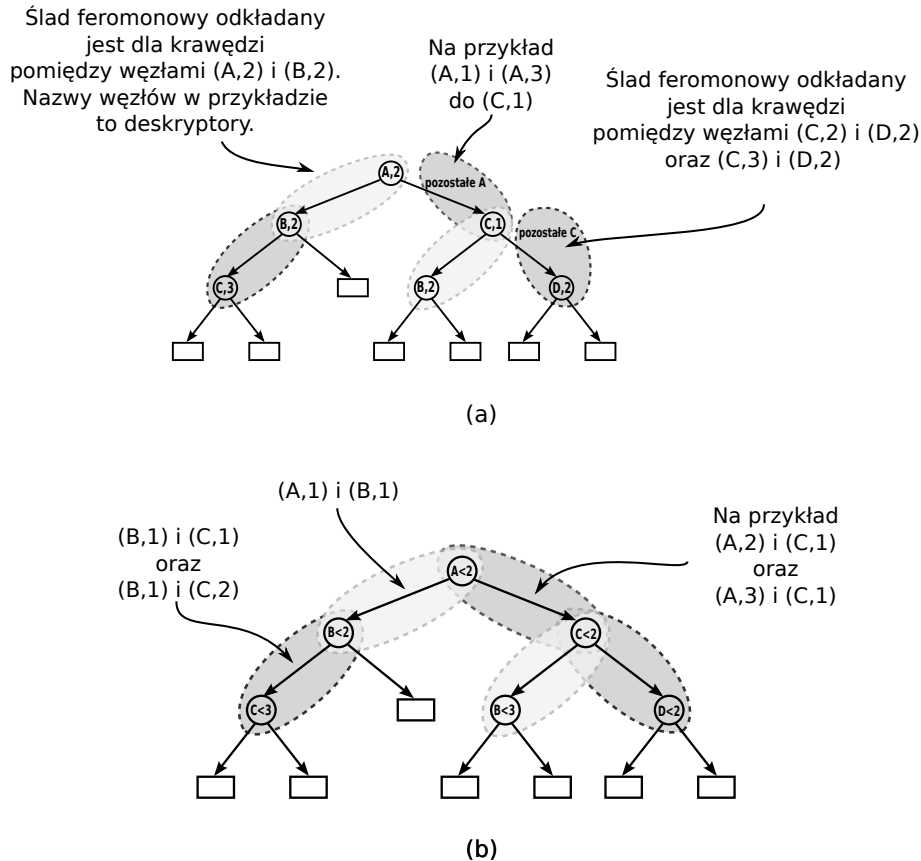
Rysunek 2: Przykład mapy feromonów powstałej po zakończeniu działania algorytmu ACDT

$$\Delta\tau_{m,m_L}(t+1) = (1 - \gamma) \cdot \tau_{m,m_L}(t) + Q(T), \quad (10)$$

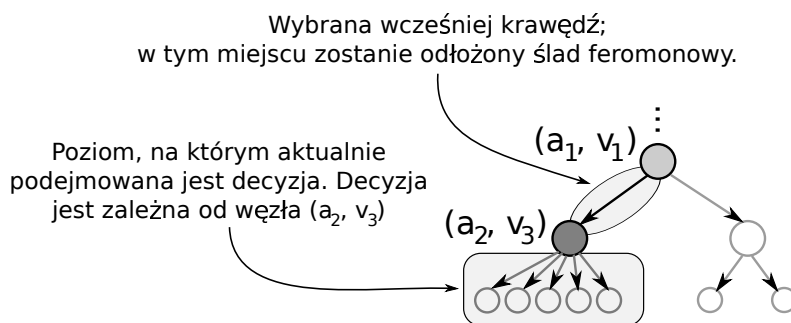
gdzie $Q(T)$ oznacza jakość drzewa decyzyjnego (2), γ jest parametrem określającym prędkość wyparowywania feromonu, natomiast t jest oznaczeniem czasu.

Budowa drzewa decyzyjnego, to wielokrotny proces decyzyjny, w którym dla każdego węzła wybierane są podziały. W przypadku algorytmu ACDT wybór podziału zależy od dotychczasowej struktury drzewa decyzyjnego (atrybutu i dopuszczalnych wartości w węźle nadrzędnym, rodzicu). Między innymi na tej podstawie wybierany jest podział w węźle (atrybut i jego wartości). W ten sposób powstaje krawędź, dla której następnie może zostać odłożony ślad feromonowy. Przykładowy sposób odkładania śladu feromonowego w algorytmie ACDT został przedstawiony na rys. 3 i 4.

W przypadku rys. 3 zaprezentowane są sposoby odkładania śladu feromonowego podczas zastosowania testu równowartościowego (a) oraz nierównowartościowego (b). W każdym przypadku ślad feromonowy odkładany jest dla każdej z krawędzi zgodnie z opisem dodanym do rysunku. Wizualizacja zaprezentowana na rys. 4 dotyczy natomiast działania algorytmu ACDT w procesie wyboru kolejnego podziału. Na rysunku wyróżniony jest węzeł, dla którego ustalony został podział oraz wybór kolejnego węzła (zależny od wcześniejszej struktury drzewa decyzyjnego).



Rysunek 3: Odkładanie śladu feromonowego w algorytmie ACDT



Rysunek 4: Przykład działania algorytmu ACDT w procesie odkładania śladu feromonowego

4.4 Rozszerzenie algorytmu ACDT

Najobszerniejsze opracowanie dotyczące algorytmu ACDT jest częścią osiągnięcia naukowego w postaci artykułu [A1]. Rozszerzenie tej pracy oraz uzupełnienie jest jednocześnie częścią 3 rozdziału monografii [M]. Prace te miały przede wszystkim pozwolić odpowiedzieć na pytanie o to, jak i dlaczego autorski algorytm ACDT działa w procesie uczenia drzew decyzyjnych. W związku z tym przeprowadzone zostały analizy zmiany macierzy śladu feromonowego (mapy feromonowe) w trakcie wykonywania algorytmu,

a także podczas kilku niezależnych jego uruchomień. W celu wykazania wkładu idei zaczerpniętych z algorytmów mrowiskowych do procesu uczenia drzew decyzyjnych zostały także wykonane badania związane ze stosowaniem śladu feromonowego oraz funkcji heurystycznej. W pracy potwierdzono zasadność stosowania śladu feromonowego oraz wyważonej heurystyki. Zauważono również, że w przypadku algorytmu ACDT dobre rezultaty uzyskiwane są podczas braku heurystyki i bazowaniu jedynie na ocenie wcześniejszych rozwiązań.

Natomiast głównym aspektem pracy [A2] było sprawdzenie różnych sposobów oceny wielkości drzew decyzyjnych oraz znalezienie kompromisu pomiędzy oceną jakości klasyfikacji a wielkością drzewa decyzyjnego. Algorytm ACDT został przeanalizowany pod względem oceny jakości rozwiązania pojedynczego agenta (jedynie część z tych obserwacji przytoczona została w podsumowaniu 3 rozdziału monografii [M]). W tym przypadku modyfikacji ulegał sposób oceny wielkości drzewa decyzyjnego, gdzie zaproponowane zostały różne warianty związane z wysokością drzewa lub liczbą jego węzłów. Algorytm ACDT służy do uczenia binarnych drzew decyzyjnych, dlatego kwestia liczby liści drzewa została pominięta, jako analogia do ogólnej liczby węzłów. Sprawdzony został również wpływ zmiany wartości parametru ϕ (wzór (2)) na uzyskiwane rezultaty. Wyniki tej pracy pozwoliły na obserwacje frontu Pareto, który tworzą rozwiązania uzyskane podczas wykonywania algorytmu ACDT.

Inny aspekt związany z funkcją oceny jakości rozwiązania (2) badany był natomiast w pracy [B3], gdzie analizie poddana została ocena jakości klasyfikatora ($ev(T, S)$). Rozszerzenie tych badań znajduje się także w rozdziale 4 monografii [M]. Rozważania oraz przeprowadzone eksperymenty dotyczyły możliwości zastosowania innych miar oceny jakości klasyfikacji (podczas oceny rozwiązania każdego agenta), przy jednoczesnym zachowaniu klasycznej heurystyki (wzór (6)). Jako testowe miary wybrane zostały popularne rozwiązania, takie jak czułość (*ang. recall*), precyzja (*ang. precision*) oraz w pewien sposób łączące je: F-miara i współczynnik korelacji Matthews'a. Natomiast ideą rozważań była możliwość uczenia klasyfikatorów dostosowanych do konkretnej miary oceny jakości klasyfikacji, w zależności od tego, do czego będzie służył zbudowany klasyfikator. Wykonane analizy pozwoliły na ciekawe obserwacje wskazujące nie tylko na to, że możliwe jest dostosowanie uczonego drzewa decyzyjnego do konkretnego celu (przykładowo większej precyzji albo czułości), ale także wielkość budowanych drzew decyzyjnych zależna jest od stosowanej miary oceny jakości klasyfikacji.

We wspomnianych pracach algorytm dostosowany był do atrybutów o wartościach ciągłych, co dokładniej przeanalizowane zostało w pracy [B8] i w pewnej części zawarte jest także w [M] (trzeci rozdział). Jest to też zasadnicze rozróżnienie w stosunku do pierwotnych koncepcji dotyczących algorytmu mrowiskowego do konstruowania drzew decyzyjnych. W trakcie opracowywania algorytmu wykonywanych było wiele analiz związanych z ustawieniami wartości parametrów (jak w większości algorytmów mrowiskowych, także w tym przypadku istnieje wiele parametrów związanych ze sterowaniem pracy algorytmu). Część z tych badań przedstawiona została na konferencji i opublikowana, jako [B7], gdzie ze szczególnym uwzględnieniem analizowano wpływ parametru q_0 na uzyskiwane rezultaty. Wspomniany parametr reguluje mechanizm eksploracji / eksploatacji w ACO. Badania te pozwoliły na oszacowanie pożądanych wartości niektórych parametrów.

Poza pracami prowadzonymi przede wszystkim przeze mnie (czasami we współpracy z innymi – zgodnie z deklaracjami wkładu pracy współautorów przytoczonymi w liście publikacji) podejmowane były próby modyfikacji algorytmu ACDT przez innych naukowców oraz adaptacji do rzeczywistych rozwiązań. Jest wiele cytowań prac związanych z algorytmem ACDT, z których wyróżnić należy pracę Otero i innych [17]. Idea jest podobna, a pseudokod identyczny jak w przypadku algorytmu ACDT, ale autorzy w proponowanym algorytmie wprowadzili znaczące modyfikacje. Przede wszystkim, jako informację heurystyczną zastosowali rozwiązania znane z algorytmu C4.5 [19]. Wpłynęło to zasadniczo na strukturę budowanego drzewa - nie jest ono binarne. Bardzo ważnym aspektem związanym z algorytmem proponowanym przez Otero i innych jest macierz feromonowa, która zachowuje informacje nie tylko o atrybutach i wartościach węzłów, ale także o poziomie drzewa decyzyjnego, na którym jest ten węzeł.

Następnie Otero i inni [20] zaproponowali rozwiązanie, w którym dla każdej z klas decyzyjnych budowane jest jedno, osobne drzewo decyzyjne. Zaproponowany model składa się z oddzielnych drzew decyzyjnych DT_1, DT_2, \dots, DT_C , gdzie C jest liczbą klas decyzyjnych. W tym przypadku DT_l jest odpowiedzialny za rozróżnienie klasy l od wszystkich pozostałych klas decyzyjnych.

Praktyczne zastosowanie algorytmu ACDT zaproponowane zostało przez Surjandari i innych [23]. Autorzy rozważali algorytm ACDT do oceny długości trwania tzw. suchego doku. Inne rozważania związane z praktycznym zastosowaniem algorytmu ACDT do klasyfikacji wiadomości elektronicznych, będące wynikiem mojej współpracy, opublikowane zostały m.in. w [DB3], [DB5] i [DB7] oraz nieco szerzej opisane w części 5.2.

Bardziej złożone modyfikacje pozwalające dostosować algorytm ACDT do praktycznych zastosowań zaproponowane zostały w pracach [B1] i [B5] oraz opisane w rozdziale 5 monografii [M]. W przypadku pracy [B1] rozważano zastosowanie uczenia drzew decyzyjnych opartych na algorytmach mrowiskowych w zakresie danych finansowych. Prace z algorytmem ACDT w procesie przewidywania zmian na rynku forex (czyli międzynarodowym rynku walutowym, par walutowych). Obecnie na rynek forex z punktu widzenia algorytmów uczenia maszynowego można patrzeć, jak na ogromny, często niezbadany jeszcze zbiór danych. Badania związane z artykułem [B1], są o tyle ciekawe, że doprowadziły do zasadniczej modyfikacji algorytmu ACDT. W tym przypadku nie jest dokonywana klasyczna klasyfikacja, ale przeprowadzane jest swego rodzaju wyszukiwanie asocjacji, czy też selekcja cech. W prezentowanym problemie, zasadniczą kwestią jest znalezienie zależności pomiędzy wskaźnikami stosowanymi w analizie rynku forex. Celem pracy zmodyfikowanego algorytmu ASC-ACDT jest znalezienie pewnego zestawu cech (wskaźników), które wystarczają do dalszej analizy rynku finansowego wykonywanej przez eksperta. W związku z tym, algorytm ASC-ACDT skonstruowany jest tak, aby dokonywać predykcji każdej z cech (na podstawie wszystkich pozostałych cech). W przypadku dobrej jakości przewidywania, można uznać, że cecha (czyli wskaźnik), która była w danym momencie uznana za atrybut decyzyjny, jest redundantna.

Problematyka poruszana w przypadku pracy [B5] nie wymagała już tak dużych zmian. W tym przypadku problemem były nierównomierne klasy decyzyjne w zbiorze danych dotyczącym wiązań wodorowych szerzej opisanych w [4]. Aby poprawić sprawność algorytmu ACDT zaproponowane zostało zastosowanie specjalnie przygotowanych zbiorów uczących — zmiennych podczas pracy algorytmu. Artykuł [B5] powiązany jest także z uczeniem zespołów klasyfikatorów opisanym w części 4.5.

4.5 Uczenie zespołów klasyfikatorów z zastosowaniem algorytmu mrowiskowego

Analiza pracy algorytmu ACDT wpłynęła na obserwacje związane z dużą różnorodnością rozwiązań będących wynikiem niezależnych uruchomień algorytmu ACDT. W szczególności dokładne badania związane ze śladem feromonowym przeprowadzone z zastosowaniem map feromonowych wpłynęły na koncepcję budowy wielu niezależnych drzew decyzyjnych, które wspólnie można połączyć w zespół klasyfikatorów. Celem takiego rozwiązania jest budowa bardziej stabilnych klasyfikatorów, które można używać do pracy z trudnymi do analizy i złożonymi zbiorami danych.

Pierwsze badania z tym związane opublikowane zostały w pracy [B6] oraz zaprezentowane są w rozdziale 7 [M]. Można powiedzieć, że dotyczą one przede wszystkim analizy różnych rozwiązań pozwalających na zastosowanie ACO w uczeniu zespołów klasyfikatorów (w szczególności na podstawie algorytmu ACDT). Generalnie rozwiązania te nazwane zostały algorytmami mrowiskowymi do budowy lasów decyzyjnych (*ang. ant colony decision forest, ACDF*) i bazowały na rozwiązaniach znanych z podstawowych zespołów klasyfikatorów (jak bagging i lasy losowe [1, 2, 3]), dostosowanych jednak do zastosowania z ACO. Dopiero w późniejszych wersjach algorytm ACDF został przystosowany do adaptacji, samoadaptacji oraz zastosowania wag dla obiektów w zbiorach danych. Rozwiązania te opublikowane są m.in. w pracach [A2] i [B4] oraz opisane w rozdziale 8 monografii [M].

Analogicznie do podejścia ACDT opisanego wzorem (1), podejście ACDF przedstawić można jako szóstkę:

$$ACDF = \langle (X, A \cup \{c\}), Z, z, ants, p_{m, m_L(i,j)}(t), S \rangle, \quad (11)$$

na którą składa się:

$(X, A \cup \{c\})$ — tabela decyzyjna reprezentująca problem, przedstawiana jako dwójka, gdzie X jest zbiorem obiektów, a A jest zbiorem atrybutów, w tym atrybut decyzyjny c ,

Z — zbiór będący rozwiązaniem problemu i zawierający z drzew decyzyjnych zbudowanych z zastosowaniem algorytmu ACDT, z zasadami zależnymi od wersji algorytmu ACDF oraz z jednakową wagą głosu (głosowanie proste),

oraz elementy systemu mrowiskowego:

$ants$ — liczba mrówek w iteracji,

$p_{m, m_L(i,j)}(t)$ — reguła wyboru rozwiązania w każdym węzle drzewa decyzyjnego w czasie t ,

S — zbiór dopuszczalnych obiektów w węzle.

Należy zauważyć, że zbiór klasyfikatorów Z zawiera z klasyfikatorów zbudowanych według zasad narzuconych przez algorytm ACDF. W większości przypadków te klasyfikatory są ze sobą powiązane i zależne od siebie. W działaniu algorytmu ACDF zasadniczą jego cechą jest ślad feromonowy, który w przypadku zespołów działa jak sprzężenie zwrotne nie tylko wobec pojedynczego klasyfikatora, ale niejednokrotnie wobec całego zbioru Z .

W tym podejściu, dla każdego klasyfikatora ze zbioru Z przekazywany jest pewien podzbiór tabeli decyzyjnej $(X, A \cup \{c\})$ zależny nie tylko od zbioru akceptowanych obiektów S , ale także od sposobu budowy podzbioru uczącego (pseudopróby) dla każdego klasyfikatora. Sposób budowy pseudopróby jest natomiast zależny od modelu przyjętego w danej wersji algorytmu ACDF.

W pracy [B6] (a także rozdziale 7 [M]) wszystkie modele algorytmu ACDF ideowo związane są z algorytmem lasów losowych zaproponowanym przez Breimana i innych [3]. Celem tej pracy było zbadanie, jak idea zapamiętywania rozwiązań poprzez ślad feromonowy sprawdzi się w połączeniu z ciągłą zmianą przestrzeni rozwiązań. Wspomniana zmiana przestrzeni rozwiązań realizowana jest poprzez zmianę analizowanych przypadków (nowe pseudopróby) oraz ograniczanie liczby cech (atrybutów). W rezultacie badaniom i teoretycznym rozważaniom podjętych zostało 7 różnych wersji algorytmu ACDF.

Wszystkie obserwacje wskazały, że możliwy jest dalszy rozwój algorytmów mrowiskowych w procesie uczenia zespołów klasyfikatorów. Kolejne analizy opisane w [B4] oraz częściowo w rozdziale 8 [M], były związane z procesem tworzenia podzbioru uczącego (pseudopróby). Dynamiczne pseudopróby tym razem tworzone były w trakcie pracy algorytmu tak, aby uzależnić je od sprawności dotychczas powstałych klasyfikatorów.

W zmodyfikowanym, adaptacyjnym algorytmie ACDF (nazwanym aACDF) początkowe prawdopodobieństwo wyboru obiektu do podzbioru uczącego wynosi $\frac{1}{n}$ (jak w przypadku baggingu, lasów losowych, czy też wcześniejszych wersji algorytmu ACDF), gdzie n jest liczbą obiektów w zbiorze. W kolejnych generacjach wartość tego prawdopodobieństwa zależy już jednak od wprowadzonej wagi tego obiektu. W przypadku nieprawidłowej klasyfikacji współczynnik ten jest zwiększany zgodnie ze wzorem:

$$we_i = \begin{cases} 1, & \text{jeśli obiekt jest dobrze sklasyfikowany} \\ 1 + \lambda \cdot n, & \text{w przeciwnym przypadku,} \end{cases} \quad (12)$$

gdzie λ to parametr wpływający na wzmocnienie dla błędnie sklasyfikowanych obiektów.

Natomiast prawdopodobieństwo wyboru obiektu jest obliczane według wzoru:

$$pp(x_i) = \frac{we(x_i)}{\sum_{j=1}^n we(x_j)}. \quad (13)$$

Należy przede wszystkim zwrócić uwagę na zastosowane wzmocnienie dla obiektów, które są gorzej sklasyfikowane przez aktualny zespół klasyfikatorów. Innymi słowy podczas tworzenia pierwszej pseudopróby wszystkie przypadki mają takie samo prawdopodobieństwo wyboru (podczas losowania ze zwracaniem), następnie, po dodaniu pierwszego klasyfikatora do zespołu, wykonywana jest aktualizacja prawdopodobieństwa wyboru (wzór (13)), w taki sposób, że przypadki, które są błędnie sklasyfikowane przez stworzony klasyfikator uzyskują swego rodzaju wzmocnienie, zgodnie ze wzorem (12). Po zbudowaniu kolejnego drzewa decyzyjnego i dodaniu go do zespołu następuje ponowna aktualizacja wag dla każdego z przypadków. Cały proces jest powtarzany aż do uzyskania odpowiedniej liczby klasyfikatorów w zespole.

Choć opisane powyżej rozwiązanie przypomina idee znane z algorytmu boosting [21] (inspirowanego przez [13]), to dopiero rozważania ujęte w pracy [A2] (zawarte też częściowo w rozdziale 8 [M]) bezpośrednio odwoływały się właśnie do boostingu. Artykuł [A2] związany jest z dwiema kolejnymi wersjami algorytmu ACDF: samoadaptacyjny (saACDF) oraz właśnie bazujący na boostingu — ACDF-Boost.

W trakcie analizy algorytmu aACDF zauważono, że większe wsparcie dla obiektów gorzej sklasyfikowanych (w trakcie uczenia zespołu klasyfikatorów) wpływa na poprawę jakości ostatecznej wersji zespołu. Jednakże zastosowanie parametru λ wymaga wcześniejszego dostrajania algorytmu. Dlatego właśnie zaproponowana została modyfikacja nazwana samoadaptacyjny ACDF (saACDF). W tym przypadku waga każdego przypadku w zbiorze uczącym jest zależna nie tylko od tego, czy został błędnie sklasyfikowany przez zespół klasyfikatorów, ale także od tego, jak dużo klasyfikatorów błędnie sklasyfikowało ten przypadek. Zostało to zawarte we wzorze:

$$we(x_i) = \begin{cases} 1, & \text{jeśli } w = 0 \\ w \cdot n & \text{w przeciwnym przypadku,} \end{cases} \quad (14)$$

gdzie w jest liczbą klasyfikatorów, przy pomocy których obiekt x_i został błędnie sklasyfikowany.

Drugi z algorytmów — ACDF-Boost, jest połączeniem algorytmu ACDF z rozwiązaniami zastosowanymi w algorytmie AdaBoost [10, 11]. Dla kolejnych populacji pseudopróby budowane są na podstawie losowania ważonego, gdzie każdy z obiektów ma przypisaną wagę. Pomimo zaczerpnięcia rozwiązań stosowanych w boostingu (czy też algorytmie AdaBoost), algorytm ACDF-Boost różni się w sposobie działania w stosunku do tego klasycznego podejścia. W przypadku ACDF-Boost klasyfikatory nie są budowane w

bezpośredni sposób na podstawie algorytmu ACDT, ale w skład lasu decyzyjnego wchodzi drzewa budowane na podstawie śladu feromonowego wcześniej powstałych drzew. Ponadto, podobnie jak w lasach losowych, drzewa decyzyjne konstruowane są na ograniczonej liczbie atrybutów.

Reasumując, w pracy [A2] zauważono, że saACDF jest metodą sekwencyjnego uczenia się, w przypadku której ważna jest aktualna postać zespołu klasyfikatorów. Nie jest to klasyczne odwzorowanie innych, sekwencyjnych metod uczenia się zespołów klasyfikatorów (jak np. boosting), ponieważ w tym przypadku prawdopodobieństwo wyboru obiektów do pseudopróby wyznaczane jest na podstawie klasyfikacji tych obiektów przez jednorodne klasyfikatory aktualnie znajdujące się w zespole. Nie mają znaczenia wyniki klasyfikacji dokonywane na wcześniejszych etapach budowy zespołu klasyfikatorów. Natomiast ACDF-Boost bazuje na całej historii budowy klasyfikatorów. Jest to metoda sekwencyjnego uczenia się, w której każdy kolejny jednorodny klasyfikator budowany jest w zależności od tego, jak radziły sobie wszystkie poprzednie klasyfikatory. Prawdopodobieństwo wyboru obiektów do pseudopróby jest zależne od każdego etapu budowy zespołu klasyfikatorów.

Podobnie, jak w przypadku algorytmu ACDT, również różne wersje algorytmu ACDF analizowane były pod względem zastosowania ich w przypadku rzeczywistych problemów. W ten sposób, w pracy [B2] rozważano algorytm aACDF, jako narzędzie wspomagające automatyczną klasyfikację wiadomości elektronicznych (problem ten jest szerzej opisany w części 5.2). Natomiast w pracy [B5] algorytm ACDF (trzy z wersji opisanych w [B6]) został zastosowany do analizy zbioru danych z nierównomierną licznością klas decyzyjnych — wspomniany w części 4.4 zbiór danych związany z wiązaniami wodorowymi.

4.6 Podsumowanie

Prace dotyczące zastosowania algorytmów mrowiskowych do uczenia drzew decyzyjnych oraz zespołów klasyfikatorów wywodzą się przede wszystkim z obserwacji klasycznych algorytmów z tego zakresu oraz prób stosowania ACO w innych problemach uczenia maszynowego. Wedle posiadanej wiedzy jestem pierwszym autorem algorytmu mrowiskowego do konstruowania drzew decyzyjnych (potwierdzające to deklaracje wkładu pracy współautorów artykułów wymienionych w tej części autoreferatu dołączone są do wniosku). Rozwiązanie to zyskało jednak uznanie i obecnie prace z tego zakresu rozwijane są przez różne ośrodki. Bardzo często autorzy nowych prac bazują na rozwiązaniach zaproponowanych w algorytmie ACDT (w różnych wersjach).

Podobnie jest w przypadku prac dotyczących algorytmów mrowiskowych do uczenia zespołów klasyfikatorów, choć w tym przypadku można znaleźć niezależne próby łączenia ACO z zespołami klasyfikatorów. Niemniej jednak rozwiązania proponowane w przypadku autorskich algorytmów ACDF (w różnych wersjach) nie są powiązane z żadnymi znanymi mi połączeniami algorytmów mrowiskowych i zespołów klasyfikatorów. Jest to tematyka ciągle rozwijana i znajdująca swe zastosowanie przede wszystkim w przypadku analiz złożonych, rzeczywistych zbiorów danych.

Problematyka i artykuły opisane w tej części autoreferatu są jedynie częścią prac związanych z algorytmem ACDT. Wybrane zostały artykuły najważniejsze z punktu widzenia różnych rozwiązań oraz dotyczące przede wszystkim problemu uczenia maszynowego. Inne prace autorskie, jak choćby związane z możliwością zrównoleglenia algorytmu ACDT, czy też heterarchii w strukturze agentów znajdują się w części 5. Pozostałe prace innych autorów odnoszące się do algorytmu ACDT znajdują się także w przeglądach literaturowych pojawiających się w [M].

5 Omówienie pozostałych osiągnięć naukowo-badawczych

Poza głównym nurtem prac omawianym w części 4 niniejszego autoreferatu prowadzę również inne badania. Wybrane publikacje związane z pozostałymi osiągnięciami naukowo-badawczymi składają się z dwóch artykułów z listy JCR ([DA1] i [DA2]), a także 8 artykułów z innych czasopism oraz konferencji, z których recenzowane artykuły w większości indeksowane są w bazach WoS lub Scopus ([DB1] — [DB8]).

Omawiane badania można podzielić na cztery grupy dotyczące odpowiednio: zastosowania uczenia maszynowego w programach grających w gry komputerowe ([DA1], [DA2] i [DB6]), automatycznej klasyfikacji wiadomości elektronicznych ([DB3], [DB5] i [DB7]), analizy wykresów giełdowych ([DB1] i [DB2]) oraz innego zastosowania algorytmów mrowiskowych w uczeniu maszynowym ([DB4] i [DB8]).

Wszystkie wymienione poniżej prace są omówione w dalszej części autoreferatu wraz z opisem mojego wkładu pracy.

[DA1] W. Wieczorek, R. Skinderowicz, J. Kozak, P. Juszczuk, A. Nowakowski. *Selected Algorithms from the 2013 Toads-and-Frogs Blitz Tournament*. ICGA Journal, tom 36, numer 4, strony 222–227, 2013.

doi.org/10.3233/ICG-2013-36405

[Deklarowany wkład: J. Kozak 20%] (MNiSW 20 punktów)

- [DA2] W. Wieczorek, R. Skinderowicz, J. Kozak, P. Juszczuk. *New Trends in Clobber Programming*. ICGA Journal, tom 34, numer 3, strony 150–158, 2011.
doi.org/10.3233/ICG-2011-34304
[Deklarowany wkład: J. Kozak 20%] (IF=1,154; MNiSW 20 punktów)
- [DB1] P. Juszczuk, J. Kozak. *Classification and preprocessing in the stock data*. BIS, BIS International Workshops, Springer LNBIP, tom 303, strony 269–281, 2017.
doi.org/10.1007/978-3-319-69023-0_23
[Deklarowany wkład: J. Kozak 30%]
- [DB2] P. Juszczuk, J. Kozak, K. Trynda. *Decision Trees on the Foreign Exchange Market*. KES, Intelligent Decision Technologies, Springer SIST, tom 57, strony 127–138, 2016.
doi.org/10.1007/978-3-319-39627-9_12
[Deklarowany wkład: J. Kozak 30%] (MNiSW 15 punktów)
- [DB3] U. Boryczka, B. Probierz, J. Kozak. *Automatic Categorization of Email into Folders by Ant Colony Decision Tree and Social Networks Analysis*. KES, Intelligent Decision Technologies, Springer International Publishing, strony 71–81, 2016.
doi.org/10.1007/978-3-319-39627-9_7
[Deklarowany wkład: J. Kozak 40%] (MNiSW 15 punktów)
- [DB4] U. Boryczka, J. Kozak. *Adaptive Ant Clustering Algorithm with Pheromone*. Intelligent Information and Database Systems ACIIDS 2016, tom 9622 of the series Lecture Notes in Computer Science, strony 117–126, 2016.
doi.org/10.1007/978-3-662-49390-8_11
[Deklarowany wkład: J. Kozak 20%] (MNiSW 15 punktów)
- [DB5] U. Boryczka, B. Probierz, J. Kozak. *A New Algorithm to Categorize E-mail Messages to Folders with Social Networks Analysis*. Computational Collective Intelligence - 7th International Conference, ICCCI 2015, Madrid, Spain, September 21-23, 2015, Proceedings, Part II, tom 9330 serii Lecture Notes in Computer Science, strony 89–98. Springer, 2015.
doi.org/10.1007/978-3-319-24306-1_9
[Deklarowany wkład: J. Kozak 40%] (MNiSW 15 punktów)
- [DB6] J. Kozak, A. Nowakowski, R. Skinderowicz, W. Wieczorek. *The 3rd Silesian University Computer Combinatorial Games Tournament*. ICGA Journal, tom 37, numer 4, strony 187–190, 2014.
doi.org/10.3233/ICG-2014-37311
[Deklarowany wkład: J. Kozak 25%] (MNiSW 10 punktów)
- [DB7] U. Boryczka, B. Probierz, J. Kozak. *An Ant Colony Optimization Algorithm for an Automatic Categorization of Email*. Computational Collective Intelligence. Technologies and Applications, ICCCI, tom 8733 serii Lecture Notes in Computer Science, strony 583–592. Springer, 2014.
doi.org/10.1007/978-3-319-11289-3_59
[Deklarowany wkład: J. Kozak 40%] (MNiSW 10 punktów)
- [DB8] U. Boryczka, J. Kozak, R. Skinderowicz. *Heterarchy in Constructing Decision Trees - Parallel ACDT*. Transactions on Computational Collective Intelligence, tom 10 serii Lecture Notes in Computer Science, strony 177–192. Springer, 2013.
doi.org/10.1007/978-3-642-38496-7_12
[Deklarowany wkład: J. Kozak 50%] (MNiSW 5 punktów)

5.1 Zastosowanie uczenia maszynowego w programach grających w gry komputerowe

Jedna z serii artykułów związanych z pozostałym osiągnięciem naukowo-badawczym dotyczy (w części przygotowanej przeze mnie) zastosowania uczenia maszynowego w programach grających w gry komputerowe, konkretniej gry kombinatoryczne. Są to trzy artykuły [DA1], [DA2] i [DB6], z których każdy dotyczy analizy różnych sposobów przygotowania programów przeznaczonych do grania w gry kombinatoryczne.

Przykładowo, w przypadku artykułu [DA2] autorska część pracy związana była z przygotowaniem algorytmu, który uczy się wykonywania ruchu w grze clobber [5] na podstawie ruchów przeciwnika.

Nieco bardziej złożone rozwiązania zastosowane były natomiast w przypadku [DB6] i przede wszystkim [DA1]. Autorski algorytm przygotowany na potrzeby gry T&F [5] użytej w artykule [DA1] związany jest z wyszukiwaniem wzorców i reguł. W przypadku zaproponowanego algorytmu proces budowy wzorców i reguł realizowany jest przed rozpoczęciem partii, natomiast zbiór uczący powinien zawierać zapis wcześniejszych rozgrywek, gier innych programów pomiędzy sobą. Mogą to być również partie algorytmów losowych, choć dla podniesienia jakości reguł lepszym rozwiązaniem jest bazowanie na zapisach gier bardziej rozbudowanych programów.

Podczas rozgrywki, decyzja związana z wykonaniem każdego ruchu ustalana jest na podstawie tego samego zbioru reguł – niezależnie od momentu gry. Wybierane są jednak tylko te reguły, które są najbardziej dopasowane do aktualnej sytuacji na planszy, a jednocześnie ruch determinowany przez nie jest możliwy do wykonania. Ostateczna decyzja ustalana jest poprzez głosowanie proste wyselekcjonowanych reguł. Program działający na podstawie algorytmu związanego z wcześniej przygotowanymi wzorcami i regułami cechuje się przede wszystkim szybkością działania. Każdy z ruchów podczas partii wykonywany jest w tym samym czasie (zależnym od wielkości zbioru uczącego). Program oparty na wzorcach i regułach przy zachowaniu bardzo szybkiego czasu działania uzyskuje poprawne wyniki, przegrywając jedynie ze znacznie bardziej złożonymi algorytmami.

Dobre wyniki algorytmów bazujących na uczeniu maszynowym w stosunku do innych analizowanych podejść uzyskiwane były przede wszystkim w przypadku wymienionych prac [DA1] i [DB6]. W tych przypadkach rozwiązania bazujące na wspomnianych, stosunkowo prostych modelach pozwalały na uzyskanie wyników zbliżonych do metod bardzo złożonych obliczeniowo.

5.2 Automatyczna klasyfikacja wiadomości elektronicznych

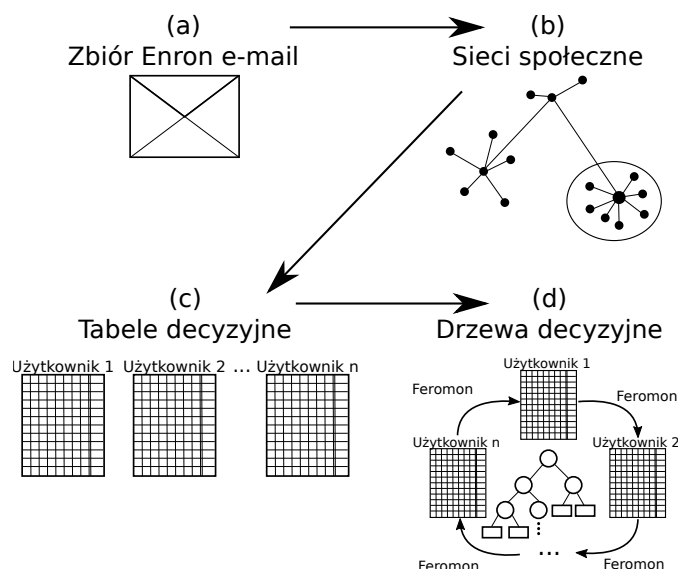
Klasyfikacja wiadomości elektronicznych jest problemem złożonym, gdyż automatyczna metoda klasyfikacji może się sprawdzić u jednego użytkownika, a u innego może prowadzić do błędów. Ponadto informacje docierają w różnym czasie, co powoduje dodatkowe trudności. Do rozwiązania tego problemu może przyczynić się analiza kontaktów pomiędzy nadawcą a odbiorcami wiadomości elektronicznych, a także drzewiasta struktura folderów utworzona w poszczególnych skrzynkach odbiorczych.

Powstało już kilka wersji algorytmu ACDT dostosowanego do rozwiązywania tego problemu. W artykułach [DB3], [DB5] i [DB7] oraz częściowo w 5 rozdziale [M] stosowane były różne wariacje oraz hybrydy algorytmu ACDT w procesie analizy zbioru wiadomości elektronicznych Enron [14].

W efekcie kolejnych rozważań, począwszy od pracy [DB7], a następnie [DB8] uzyskano hybrydę, której schemat działania przedstawiony jest na rys. 5. Pierwszym krokiem (a) jest stworzenie sieci społecznej [22] opartej na kontaktach między nadawcą oraz odbiorcą wiadomości uzyskanych z zestawu danych Enron. Następnie podczas analizy tej sieci wybrani zostają kluczowi użytkownicy (węzły) i ich najbliżsi sąsiedzi (b). Kolejnym krokiem, który należy podjąć w celu wdrożenia proponowanego algorytmu, jest przekształcenie zestawu danych w tabelę decyzyjną, osobno dla każdej skrzynki pocztowej w danej grupie (c). Ostatecznie podczas uczenia drzewa decyzyjnego przeznaczonego dla każdego użytkownika analizowana jest sieć komunikacyjna grupy przypisanej do tego użytkownika. Klasyfikator jest budowany dla wybranego użytkownika za pomocą algorytmu, a następnie konstruowany jest klasyfikator dla każdej kolejnej osoby w grupie przy użyciu tej samej macierzy śladu feromonów. Po zbudowaniu klasyfikatorów i ustabilizowaniu macierzy śladu feromonowego powstaje ostateczny klasyfikator dla danego użytkownika kluczowego (d). Podczas uczenia tego klasyfikatora zachowana jest informacja związana z decyzjami podejmowanymi przez pozostałych członków grupy (za pośrednictwem śladu feromonowego).

5.3 Analiza wykresów giełdowych

Znajomość mechanizmów uczenia maszynowego pozwoliły na rozwijanie współpracy ze specjalistami innych dziedzin naukowych. Dzięki temu powstało kilka prac, z których dwie [DB1] oraz [DB2] oznaczone zostały jako pozostałe osiągnięcia naukowe. Obydwie prace związane były z zastosowaniem algorytmów uczenia maszynowego w predykcji skupionej na rynku walutowym. W szczególności analizowana była skuteczność algorytmów bazujących na drzewach decyzyjnych w problemie predykcji wartości instrumentów finansowych (par walutowych w pracach [DB1] i [DB2] oraz indeksów giełdowych w pracy [DB2]). W efekcie przeprowadzonych analiz zaproponowany został między innymi nowatorski mechanizm oceny jakości predykcji uwzględniający bezwzględną zmianę wartości ceny instrumentu w badanym oknie czasowym.



Rysunek 5: Diagram pracy hybrydowego algorytmu ACDT w problemie klasyfikacji wiadomości elektronicznych

5.4 Inne zastosowanie algorytmów mrowiskowych w uczeniu maszynowym

W grupie pozostałych osiągnięć umieszczona jest także praca powiązana z zastosowaniem algorytmów mrowiskowych w uczeniu drzew decyzyjnych, jednak jej cel odbiega od tematu podstawowego osiągnięcia naukowo-badawczego. W pracy [DB8] podjęto udaną próbę zrównoleglenia algorytmu ACDT. Podczas prowadzonych badań analizowano przyspieszenie związane z równoległym algorytmem ACDT, a także różne kanały komunikacji pomiędzy agentami. Zaproponowane strategie kooperacji między koloniami mogą być zaliczone do grona algorytmów heterarchicznych. Składają się z agentów o takim samym stopniu ważności (nie istnieje hierarchia ważności agentów), a komunikacja oparta jest na wspomnianych kanałach komunikacji. Współpraca pomiędzy agentami w koloniach odbywała się na dwóch poziomach: wewnątrz kolonii - stosując ślad feromonowy i pomiędzy grupami - wykorzystując jawną wymianę informacji w postaci znalezionych, najlepszych rozwiązań. Zaproponowano dwie strategie współpracy między koloniami i przetestowano je pod kątem dokładności klasyfikacji i wydajności obliczeniowej.

Innym podejściem związanym z zastosowaniem uczenia maszynowego oraz algorytmów mrowiskowych było zaproponowanie w pracy [DB4] modyfikacji algorytmu do klastrowania. Celem tych badań było opracowanie i zbadanie algorytmu mrowiskowego do klastrowania z zastosowaniem macierzy śladów feromonowych.

Literatura

- [1] L. Breiman. Bagging predictors. Technical Report 421, Department of Statistics, University of California at Berkeley, September 1994.
- [2] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
- [3] L. Breiman. Random forests. *Mach. Learn.*, 45:5–32, October 2001.
- [4] I. Chikalov, P. Yao, M. Moshkov, and J. C. Latombe. Learning probabilistic models of hydrogen bond stability from molecular dynamics simulation trajectories. *BMC Bioinformatics*, 12(S-1):S34, 2011.
- [5] E. D. Demaine, M. L. Demaine, and R. Fleischer. Solitaire clobber. *Theoretical Computer Science*, 313(3):325–338, 2004.
- [6] M. Dorigo. *Optimization, Learning and Natural Algorithms (in Italian)*. PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, IT, 1992.
- [7] M. Dorigo, G. Di Caro, and L. M. Gambardella. Ant algorithms for distributed discrete optimization. *Artif. Life*, 5(2):137–172, 1999.

- [8] M. Dorigo and L. M. Gambardella. A study of some properties of Ant-Q. In *Proceedings of Fourth International Conference on Parallel Problem Solving from Nature, PPSNIV*, pages 656–665, Berlin, 1996. Springer-Verlag.
- [9] M. Dorigo, V. Maniezzo, and A. Colorni. Positive feedback as a search strategy. Technical Report 91–016, Politecnico di Milano, Italy, 1991.
- [10] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pages 148–156, 1996.
- [11] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.
- [12] L. Hyafil and R. L. Rivest. Constructing optimal binary decision trees is np-complete. *Inf. Process. Lett.*, 5(1):15–17, 1976.
- [13] M. Kearns. Thoughts on hypothesis boosting. Project for Ron Rivest’s machine learning course at MIT., 1988.
- [14] B. Klimt and Y. Yang. The enron corpus: A new dataset for email classification research. In *European Conference on Machine Learning*, pages 217–226. Springer, 2004.
- [15] J. Koronacki and J. Cwik. *Statystyczne systemy uczace sie*. Exit, Warsaw, 2008.
- [16] O. J. Murphy and R. L. McCraw. Designing storage efficient decision trees. *IEEE Trans. Computers*, 40(3):315–320, 1991.
- [17] F. E. B. Otero, A. A. Freitas, and C. G. Johnson. Inducing decision trees with an ant colony optimization algorithm. *Appl. Soft Comput.*, 12(11):3615–3626, 2012.
- [18] R. S. Parpinelli, H. S. Lopes, and A. A. Freitas. An ant colony algorithm for classification rule discovery. In H. Abbas, R. Sarker, and C. Newton, editors, *Data Mining: a Heuristic Approach*, pages 191–208. Idea Group Publishing, London, 2002.
- [19] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
- [20] K. M. Salama and F. E. B. Otero. Learning multi-tree classification models with ant colony optimization. In *IJCCI (ECTA)*, pages 38–48, 2014.
- [21] R. E. Schapire. The strength of weak learnability. *Machine Learning*, 5:197–227, 1990.
- [22] J. Scott. *Social network analysis*. Sage, 2017.
- [23] I. Surjandari, A. Dhini, A. Rachman, and R. Novita. Estimation of dry docking duration using a numerical ant colony decision tree. *International Journal of Applied Management Science*, 7(2):164–175, 2015.