

# Wybrane technologie eksploracji i reprezentacji wiedzy

pod redakcją naukową Zenona A. Sosnowskiego



# **WYBRANE TECHNOLOGIE EKSPLOKACJI I REPREZENTACJI WIEDZY**

pod redakcją naukową  
Zenona A. Sosnowskiego



OFICyna WYDAWNICZA POLITECHNIKI BIAŁOSTOCKIEJ  
BIAŁYSTOK 2022

Recenzenci:  
prof. dr hab. inż. Janusz Kacprzyk  
dr hab. Robert Milewski

Redaktor naukowy dyscypliny informatyka techniczna i telekomunikacja:  
prof. dr hab. Jarosław Stepaniuk

Korekta językowa:  
mgr Agnieszka Polecka (język polski)  
mgr Maciej Pacholski (język angielski)

Skład, grafika i okładka:  
Marcin Dominów

© Copyright by Politechnika Białostocka, Białystok 2022

ISBN 978-83-67185-36-3  
ISBN 978-83-67185-37-0 (eBook)  
DOI: 10.24427/978-83-67185-37-0



Publikacja jest udostępniona na licencji  
Creative Commons Uznanie autorstwa-Użycie niekomercyjne-Bez utworów zależnych 4.0  
(CC BY-NC-ND 4.0).

Pełną treść licencji udostępniono na stronie  
[creativecommons.org/licenses/by-nc-nd/4.0/legalcode.pl](https://creativecommons.org/licenses/by-nc-nd/4.0/legalcode.pl).  
Publikacja jest dostępna w Internecie na stronie Oficyny Wydawniczej PB.

Druk: PPH Remigraf sp. z o.o.

---

Oficina Wydawnicza Politechniki Białostockiej  
ul. Wiejska 45C, 15-351 Białystok  
e-mail: [oficina.wydawnicza@pb.edu.pl](mailto:oficina.wydawnicza@pb.edu.pl)  
[www.pb.edu.pl](http://www.pb.edu.pl)

# Spis treści

Wstęp .....	5
Rozdział I	
Paradygmat Turinga	
<i>Kazimierz Trzęsicki</i> .....	7
Rozdział II	
Modele klasyfikacyjne oparte na wielowymiarowych wektorach cech	
<i>Leon Bobrowski</i> .....	43
Rozdział III	
Predicting the readmission risk of heart failure patients based on claims data	
<i>Tomasz Łukaszuk, Jerzy Krawczuk, Mariusz Ferenc</i> .....	63
Rozdział IV	
Alternatywny algorytm dla transformaty Hough'a	
wyszukujący wzorce liniowe	
<i>Paweł Zabielski</i> .....	79
Rozdział V	
Image processing methods in semi-automatic	
mobile application testing framework	
<i>Maciej Szymkowski, Tomasz Łukaszuk, Andrzej Chmielewski</i> .....	87
Rozdział VI	
Dlaczego nazwy liczb odczytujemy odwrotnie? Problem reprezentacji liczb	
<i>Kazimierz Trzęsicki</i> .....	101
Spis tabel .....	119
Spis rysunków .....	121



# Wstęp

Odpowiedzią na gwałtowny wzrost ilości informacji gromadzonych w bazach i hurtowniach danych jest intensywnie rozwijająca się dziedzina odkrywania wiedzy w bazach danych (*Knowledge Discovery in Databases*) oraz eksploracji danych (*Data Mining*). Techniki eksploracji danych pozwalają na znajdowanie wcześniej nieznanymi zależności i schematów, które mogą być wykorzystane do wspomaganie podejmowania decyzji lub opisu danych.

Monografia prezentuje wybrane zagadnienia związane z reprezentacją i odkrywaniem wiedzy postrzeganym jako złożony proces selekcji i transformacji danych, ich eksploracji, a następnie interpretacji uzyskanych wyników. Autorzy rozdziałów niniejszej książki prezentują szeroki zakres tematyczny wybranych metod eksploracji i reprezentacji wiedzy.

Rozdział pierwszy Kazimierza Trzęsickiego PARADYGMAT TURINGA jest poświęcony przedstawieniu początków paradygmatu Turinga oraz niektórych problemów, jakie w jego ramach są formułowane i rozwiązywane. Zgodnie z tym paradygmatem właściwym językiem wiedzy jest język algorytmiki.

W rozdziale drugim MODELE KLASYFIKACYJNE OPARTE NA WIELOWYMIAROWYCH WEKTORACH CECH, autorstwa Leona Bobrowskiego, przedstawiono zagadnienia dotyczące możliwości projektowania klasyfikatorów na podstawie zbiorów danych zawierających małą liczbę wysokowymiarowych wektorów cech. Badane są teoretyczne właściwości warstw złożonych z dużej liczby klasyfikatorów liniowych opartych na wyselekcjonowanych podzbiorach cech o małych liczebnościach.

Tomasz Łukaszuk, Jerzy Krawczuk oraz Mariusz Ferenc, w rozdziale trzecim PRZEWIDYWANIE RYZYKA READMISJI PACJENTÓW Z NIEWYDOLNOŚCIĄ SERCA NA PODSTAWIE DANYCH Z CLEIM'ÓW, przedstawiają w jaki sposób można przeprowadzić estymację ryzyka readmisji i jakich wyników można się spodziewać dla dużej liczby epizodów niewydolności serca. Problem został przeanalizowany w badaniu empirycznym z wykorzystaniem czterech klasyfikatorów i kilku metod kodowania kodów diagnostycznych.

W rozdziale czwartym ALTERNATYWNY ALGORYTM DLA TRANSFORMATY HOUGH'A WYSZUKUJĄCY WZORCE LINIOWE, Paweł Zabielski opisuje problem wydobywania wzorców liniowych z obrazów 2D. Do rozważenia są również wzorce bliskie wzorcom liniowym – wzorce kolinearne. Zastosowana została tu metoda minimalizacji wypukłych i odcinkowo-liniowych funkcji kryterialnych. Zaprezentowana metoda stanowi alternatywę dla popularnej transformacji Hough'a.

Rozdział piąty Macieja Szymkowskiego, Tomasza Łukaszuka i Andrzeja Chmielewskiego, METODY PRZETWARZANIA OBRAZÓW WE FRAMEWORK'U DO PÓŁAUTOMATYCZNEGO TESTOWANIA APLIKACJI MOBILNYCH, proponuje koncepcyjne podejście do półautomatycznego testowania aplikacji mobilnych. W rozdziale dokładnie opisano, jakie metody przetwarzania obrazu mogą być wykorzystane w takim rozwiązaniu. Rozważano metody segmentacji i dopasowywania szablonów i ich użycie w reprodukcji testów na różnych urządzeniach.

W rozdziale szóstym DLACZEGO NAZWY LICZB ODCZYTUJEMY ODWROT-NIE? PROBLEM REPREZENTACJI LICZB, autorstwa Kazimierza Trzęsickiego, przedstawiono kwestie reprezentacji liczb. W szczególności początki liczenia, ideę „zera” oraz zapisu liczb.

Zespół autorski niniejszej monografii ma nadzieję, że treści w niej zawarte zainteresują potencjalnych czytelników. Wiele osób pomogło w przygotowaniu i publikacji tej książki. Chciałbym podziękować Autorom poszczególnych rozdziałów, Recenzentom za pomoc w procesie selekcji materiału, a Wydziałowi Informatyki PB za wspieranie naszych wysiłków.

Zenon A. Sosnowski  
Białystok, maj 2022 r.

# Rozdział I

## Paradygmat Turinga

Kazimierz Trzęsicki  
Uniwersytet w Białymstoku

*Tolle numerum omnibus rebus et omnia pereunt  
Odbierz wszystkiemu liczbę, a wszystko przepadnie.*

św. Izydor z Sewilli, patron Internetu

**Streszczenie:** Rozwój nauki dokonuje się nieliniowo, a poszczególne etapy wyróżnia charakterystyczny paradygmat badań naukowych. Według paradygmatu Galileusza właściwym językiem wiedzy jest język matematyki. Według paradygmatu Turinga właściwym językiem wiedzy jest język algorytmiki. Wskazujemy źródła i początki paradygmatu Turinga oraz niektóre problemy, jakie w jego ramach są formułowane i rozwiązywane.

**Słowa kluczowe:** *Galileo Galilei, Alan Turing, Konrad, Zuse, zero, cyfra arabska, paradygmat, matematyka, algorytmika*

## 1. Wstęp

Nauka uprawiana jest zgodnie z jakimś wzorcem, paradygmatem. Paradygmat jest historycznie zmienny. Miejsce jednego paradygmatu zajmuje ten, który umożliwia pełniejsze rozumienie i lepszy opis pozyskanych informacji.

Paradygmat Arystotelesa wiedzy naturalnej został zastąpiony przez paradygmat, który zwykliśmy wiązać z Galileuszem. Według tego paradygmatu tworzona była współczesna nauka (*science*). Wyznaczony przez Arystotelesa paradygmat logiki trwał do czasu Gottloba Frege. Grecki paradygmat matematyki został zastąpiony przez paradygmat, który możemy wiązać z Kartezjuszem. Czy paradygmaty współczesnej nauki nie mają charakteru historycznego, czy dalsze badania nie doprowadzą do nowych wzorców uprawiania nauki? Filozofia informatyczna postuluje nowy paradygmat, który nazywamy paradygmatem Turinga. Paradygmat Turinga zdaje się lepiej i pełniej ujmować wiedzę o obszarach, w których dominuje paradygmat Galileusza, ale i o obszarach, w których paradygmat Galileusza napotyka różnorodne ograniczenia.

Kluczowymi pojęciami filozofii informatycznej są pojęcia informacji, algorytmu i sztucznej inteligencji. Gdybyśmy mieli krótko scharakteryzować epokę informatyczną, w której przyszło nam żyć wystarczyłyby trzy terminy: informacja, algorytm, sztuczna inteligencja.



## 2. Zapis binarny

Idea kodu binarnego ma długą historię [71,98]. Leibniz, tworząc swój system binarny, wskazywał poprzednika w osobie trzynastowiecznego matematyka arabskiego Abdallaha Beidhawya.

Około 1600 r. binarną notacją stosował angielski astronom Thomas Harriot. O jego osiągnięciach pisze John Shirley [85]:

*Though it is frequently stated that binary numeration was first formally proposed by Leibniz as an illustration of his dualistic philosophy, the mathematical papers of Thomas Harriot (1560–1621) show clearly that Harriot not only experimented with number systems, but also understood clearly the theory and practice of binary numeration nearly a century before Leibniz's time.*

Chociaż często stwierdza się, że system binarny po raz pierwszy formalnie zaproponował Leibniz jako zobrazowanie swojej dualistycznej filozofii, matematyczne teksty Thomasa Harriota (1560–1621) jasno pokazują, że Harriot nie tylko eksperymentował z systemami liczbowymi, lecz także rozumiał jasno teorię i praktykę binarnej numeracji blisko na wiek przed czasami Leibniza.

Podobną opinię ma Robert Ineichen [51]:

*He is probably the first inventor of the binary system, as several manuscripts in his legacy show. In the binary system, he uses the numerals 0 and 1 and shows examples of how to move from the decimal system to the binary system and vice versa (conversio or reductio). Using further examples, he demonstrates the basic arithmetic operations.*

Prawdopodobnie jest on [Harriot] pierwszym pomysłodawcą binarnego systemu, jak ukazuje to szereg pozostawionych przez niego manuskryptów. W systemie binarnym używa numerałów 0 i 1, i podaje przykłady jak przejść z systemu dziesiętnego do systemu binarnego i *vice versa* (*conversio* lub *reductio*). Podając dalsze przykłady, demonstruje podstawowe operacje arytmetyczne.

Ineichen, jako pierwszy tekst na temat systemu binarnego wskazuje opublikowane w 1670 r. dwutomowe dzieło *Mathesis biceps vetus et nova* (1670) autorstwa Juana Caramuela y Lobkowitz (Ioannis Caramuelis). W związku z tymi pracami Harriota i Caramuela stawia się pytanie, czy Leibniz dokonał plagiatu. Na pytanie to udziela się odpowiedzi pozytywnej [1].

Pierwszego binarnego zakodowania znaków alfanumerycznych dokonał Giuseppe Peano. W latach 1887–1901 zaprojektował abstrakcyjną maszynę stenograficzną opartą na kodowaniu binarnym wszystkich sylab języka włoskiego. Razem z fonemami za pomocą 16 bitów (miał więc 65 536 kombinacji), zakodowane było 25 liter alfabetu (włoskiego) i 10 cyfr. Kod Peany nie został zauważony i był zapomniany.

Użycie kodu binarnego nie było oczywiste. Ukończony latem 1946 r. amerykański ENIAC, inaczej niż kodowane binarnie Z3, ABC i Colossus, bazował na arytmetyce dziesiętnej.

O korzystaniu w komputerach z systemu binarnego ostatecznie przesądził Burk-Goldstine-Von Neuman Report [14] z 1947 r., w którym czytamy:

*An additional point that deserves emphasis is this: An important part of the machine is not arithmetical, but logical in nature. Now logics, being a yes-no system, is fundamentally binary. Therefore, a binary arrangement of the arithmetical organs contributes very significantly towards a more homogeneous machine, which can be better integrated and is more efficient.*

*Dodatkowy punkt, który zasługuje na podkreślenie jest następujący: Ważna część maszyny nie jest natury arytmetycznej, lecz logicznej. Obecnie logika, będąc systemem tak-nie, jest zasadniczo binarna. Dlatego binarne zorganizowanie urządzeń arytmetycznych znacząco wpływa na większą homogeniczność maszyny, która może być lepiej zintegrowana i jest bardziej wydajna.*

### 3. Świat zbudowany z liczb

*Najpiękniejszy jest przedmiot którego nie ma.*

Zbigniew Herbert, Studium przedmiotu

Idea liczby jako zasady świata ma swego protagonistę w osobie Pitagorasa, który głosił: Liczba jest zasadą, źródłem i korzeniem wszystkich rzeczy [40]. Twierdził, że każdej istniejącej rzeczy odpowiada wartość numeryczna, a w średniowieczu wyrażało to scholastyczne: *dictum omne ens est scibile* (wszystkie byty są poznawalne)<sup>1</sup> [21]. Taka koncepcja liczby jako zasady świata znajduje nowe skojarzenia, kiedy pojawia się idea zera.

W styczniu 1697 r. Leibniz wraz z życzeniami urodzinowymi do swego protektora księcia Rudolfa Augusta z Brunszwika (Herzog von Braunschweig-Wolfenbüttel Rudolph August) przesłał list<sup>2</sup>, w którym omawia system binarny i ideę stworzenia z 0 jako nicością i 1 jako Bogiem [90].

Dla Leibniza [65] nicość i ciemność odpowiadają zeru, zaś promieniujący duch Boga odpowiada jedynce. Uważał bowiem, że wszystkie kombinacje powstają z jedności i nicości, co jest podobne temu, gdy mówi się, że Bóg uczynił wszystko z niczego

<sup>1</sup> Two interesting Arguments for God: Intelligibility and Desire, 2012; <http://shamelesspopery.com/two-interesting-arguments-for-god-intelligibility-desire/> [20.10.2020].

<sup>2</sup> [https://www.hs-augsburg.de/~harsch/germanica/Chronologie/17Jh/Leibniz/lei\\_bina.html](https://www.hs-augsburg.de/~harsch/germanica/Chronologie/17Jh/Leibniz/lei_bina.html) [20.03.2020].

i że były tylko dwie zasady: Bóg i nicłość. Zaprojektował medal, którego motywem przewodnim było imago creationis i ex nihil ducendis Sufficit Unum. Jedyńce odpowiada Słońce, które promieniuje na bezkształtną Ziemię, zero.

Koncepcja, że wszystko jest stworzone z 0 i 1 jest powodem, dla którego jeden z twórców algorytmicznej teorii informacji Gregory Chaitin – jak pisze nie całkiem na serio – proponuje nazwać podstawową jednostkę informacji nie „bit”, lecz „leibniz” [19, 97]:

*all of information theory derives from Leibniz, for he was the first to emphasize the creative combinatorial potential of the 0 and 1 bit, and how everything can be built up from this one elemental choice, from these two elemental possibilities. So, perhaps not entirely seriously, I should propose changing the name of the unit of information from the bit to the Leibniz!*

*cała teoria informacji wywodzi się z Leibniza, ponieważ on pierwszy podkreślił kreatywny kombinatoryczny potencjał bitu 0 i 1, i jak wszystko może być zbudowane przez ten jeden elementarny wybór, z tych dwu elementarnych możliwości. Tak, być może nie całkiem na serio, powinienem zaproponować zmianę nazwy jednostki informacji z bit na leibniz!*

Jednostka „leibniz” mogłaby być jednostką (parcel), o której pisał Hobbes. Leibniz był przekonany, że świat urządzony jest zgodnie z zasadami matematyki. Myśl tę skrótowo wyraża zdanie ([66], s. 191)<sup>3</sup>:

*Cum Deus calculat et cogitationem exercet, fit mundus*

*Gdy Bóg przemyśliwa rzeczy i rachuje, staje się świat.*

Matematyka jest narzędziem Konstruktora świata, a liczby są materiałem, z którego świat jest stworzony.

Współcześnie ideę świata jako stworzonego z przedmiotów matematycznych, Mathematical Universe Hypothesis, głosi kosmolog Max Tegmark [93, 94]. Przedmioty matematyczne istnieją w ‘platońskim niebie’. Dla wszechświata są bardziej podstawowe niż atomy i elektrony.

## 4. Nowoczesne przyrodoznawstwo

Idea matematyczności świata legła u podstaw nowoczesnego przyrodoznawstwa, a początki zwykło wiązać się z wystąpieniem Galileusza, który głosił, że księga natury zapisana jest językiem matematyki.

---

<sup>3</sup> Więcej na temat tego zapisu na marginesie rozprawki Dialogus [66] zob.[56].

Ukształtowanie nowoczesnego paradygmatu nauki w zakresie tego, co nazywano wówczas „filozofią naturalną”, w istocie było wskrzeszeniem koncepcji Archimedesesa [44]. Ta idea trwa też w średniowieczu. Roger Bacon (ur. ok. 1214, zm. 1292) w *Opus Majus* (2010) podkreśla, że:

*Et sunt quatuor scientiae magnae, sine quibus caeterae scientiae sciri non possunt, nec rerum notitia haberi: quibus scitis, potest quilibet gloriose proficere in sapientiae potestate sine difficultate et labore, non solum in scientiis humanis, sed divina. Et cuiuslibet istarum tangetur virtus non solum propter sapientiam absolute, sed respectu caeterorum praedictorum. Et harum scientiarum porta et clavis est mathematica, quam sancti a principio mundi invenerunt, ut ostendam, et quae semper fuit in usu omnium sanctorum et sapientum prae omnibus aliis scientiis. Cuius negligentia jam per triginta vel quadraginta annos destruxit totum studium Latinorum. Quoniam qui ignorat cam non potest scire caeteras scientias nec res huius mundi, ut probabo. Et, quod peius est, homines eam ignorantes non percipiunt suam ignorantiam, et ideo remedium non quaerunt. Ac per contrarium huius scientiae notitia, praeparat animum et elevat ad omnium certificatam cognitionem, ut si radices sapientiae datas circa illam cognoscat, et eas radices recte applicet ad caeterarum scientiarum et rerum cognitiones, tunc omnia sequentia poterit scire sine errore et sine dubitatione, ac de facili et potenter.*

Są cztery wielkie nauki, bez których inne nie mogą być poznane i nie można pojąć sensu rzeczy. A kiedy są znane, wówczas osiągnie się mądrość bez trudu i pracy, nie tylko w naukach człowieka, lecz również w boskich. A możliwości każdej z tych nauk ujawniają się nie tylko ze względu na samą mądrość, lecz także w stosunku do wyżej powiedzianego. A bramą i kluczem do tych nauk jest matematyka, której, jak pokażę, inwencja jest dziełem Świętego na początku świata, i która zawsze była stosowana przez wszystkich świętych i mędrców przed wszystkimi innymi naukami. Jej zaniedbanie przez trzydzieści lub czterdzieści lat doprowadziło do destrukcji całego Studium Latinorum. Udowodnię, że ci, którzy nie znają matematyki, nie mogą znać innych nauk i spraw tego świata. A co gorsza, ludzie, którzy jej nie znają, nie mają poczucia własnej ignorancji i dlatego nie szukają na nią remedium. Przeciwnie, ci którzy mają tej wiedzy znajomość przygotowują duszę i wynoszą ją do wszelkiej rzetelnej wiedzy, dzięki czemu mogą poznawać bez błędu i wątpliwości, lekko i rzetelnie.

Galileusz uzasadnia heliocentryzm, powołując się na egzegezę Biblii opartej na doktrynie św. Augustyna, w szczególności jego *De Genesi ad litteram*<sup>4</sup>[86]. W tej tradycji, której wyraźne określenie znajdujemy u Galileusza, księga natury powinna być czytana raczej za pomocą narzędzi matematycznych, aniżeli tych, które ma filozofia

<sup>4</sup> <http://www.inters.org/galilei-madame-christina-Lorraine> [02.04.2019].

scholastyczna. Księga przyrody została napisana językiem matematyki, zatem musi być interpretowana przez matematyków, a nie przez teologów. Księga natury jako matematyka zawiera niepodlegające dyskusji prawdy.

Galileusz [38] głosi, że:

*Filozofia [tj. fizyka] jest zapisana w tej wielkiej księdze – mówię o wszechświecie – która stale stoi przed naszym wzrokiem, lecz nie może być zrozumiana dopóki ktoś pierwszy nie nauczy się rozumieć języka i interpretować znaków, którymi jest napisana. Jest napisana językiem matematyki, a jej znakami są trójkąty, okręgi i inne figury geometryczne, bez których po ludzku nie jest możliwe zrozumienie jednego słowa z niej; bez tego, wędruje się wkoło w ciemnościach labiryntu.*

Galileusz powie, żeby się uczyć języka matematyki, bo jest językiem, którym mówi Bóg [116, 88]. Hall [41,42] twierdzi, że:

*nikt przedtem nie rozszerzył matematycznych metod rozumowania na ruchy rzeczywistych ciał i nikt nie ośmielił się głosić, że metoda ta była słuszna w całej fizyce, że była to jedyna słuszna metoda.*

Dodajmy, że Galileusz postrzega jeszcze matematyczność przyrody jako jej geometryczność – taka była tradycja pitagorejska. To zmieni dopiero Kartezjusz, algebraizując geometrię. Dopiero po Kartezjuszu odrzucono, głoszoną w *Analitykach* wtórych przez Arystotelesa tezę, że arytmetycznie nie można dowieść prawd geometrycznych. Po Kartezjuszu matematyka – co miało istotne znaczenie dla rozwoju nauki – stała się wiedzą o funkcjach i operacjach, a nie tylko jak wcześniej, o liczbach.

Kartezjusz jako chorowite dziecko miał przywilej późnego wstawania. Ten zwyczaj zachował jako osoba dorosła. Wykorzystał to współczesny Kartezjuszowi niemiecki filozof Daniel Lipstropius i wymyślił opowiadanie, jak Kartezjusz powziął ideę tego, co nazywamy kartezjańskim układem współrzędnych [75]. Na tę wspaniałą, rewolucyjną dla matematyki ideę, miał Kartezjusz – według tej opowiadki – wpaść wpatrzony w pełzające po suficie muchy w swojej sypialni w La Flèche w 1636 r. Zauważył mianowicie, że pozycję muchy można jednoznacznie określić przez jej odległości od ścian.

Newton tworzy rachunek różniczkowy i całkowy, bo jest to język, którym napisana jest księga natury. Rachunek różniczkowy i całkowy tworzy również Leibniz. Na marginesie dodajmy, że Newton zarzucił mu plagiat. Tak się jednak składa, że Leibniz, geniusz tworzenia symboli [75] – mając większe zrozumienie dla wyboru języka – nadał swojej wersji taką reprezentację językową, która zaowocowała rozwojem, co nie powiodło się w przypadku ujęcia proponowanego przez Newtona. Charles Babbage, twórca mechanicznego, pierwszego programowalnego komputera, dostrzegając w zakresie rachunku opóźnienie matematyki angielskiej w stosunku do francuskiej, podjął się tłumaczenia tekstów francuskich z matematyki [99]. Pisał [5,6]:

*Under these circumstances it was not surprising that I should perceive and be penetrated with the superior power of the notation of Leibniz.*

*W tych okolicznościach nie było nic zaskakującego, że powinienem pojąć i rozważyć z najwyższą uwagą notację Leibniza.*

Dla Izaaka Newtona i innych filozofów tego okresu matematyczne wyrażenie filozoficznych koncepcji obejmowało również naturalne ludzkie relacje: te same prawa poruszały fizyczną i duchową rzeczywistość. Dla ludzkich zachowań wskazywano modele matematyczne. W przypadku Pascala jest to np. słynny zakład: racjonalna osoba powinna żyć, jak gdyby Bóg istniał. Jeśli Bóg nie istnieje, to osoba ta ma skończone straty (jakieś przyjemności, luksus itp.), a jeśli istnieje, zyskać może nieskończone wiele (nieskończone szczęśliwe życie w niebie) i uniknąć nieskończonych strat (wieczność w piekle). Zakład jest pierwszym przykładem formalnego wykorzystania teorii decyzji.

Gottfried Leibniz [64,65] matematycznie modeluje stworzenie i tworzenie świata [98, 99, 101]. Za Hobbesem głosi koncepcję myślenia jako rachunku: *cogitatio est calculatio* [66]. To wszystko jest spójne z koncepcją Boga jako tego, który rachując stwarza świat. Matematyka jest narzędziem Konstruktora świata, a liczby są tworzywem, z którego świat jest uformowany. Jest to Bóg, którego logika jest taka sama jak człowieka.

Zdaniem Johannesesa Keplera również anioły poruszają planety zgodnie z modelem matematycznym.

Ideę Boga (Boga Spinozy) jako „matematyka” głosi Einstein [52]:

*God does not care about our mathematical difficulties. He integrates empirically.*

*Bóg nie martwi się naszymi matematycznymi trudnościami. On całkuje empirycznie.*

Jest to [45]:

*Fundamentalna hipoteza, przyjmowana milcząco w samej metodzie współczesnych zmatematyzowanych nauk empirycznych [która] głosi, że w materialnym świecie nie ma niczego, czego nie dałoby się opisać matematycznie.*

Poszerzenie idei matematyczności natury na inne dziedziny głosiło wielu, np. Nicolas de Condorcet mówił o zastosowaniu rachunku różniczkowego i całkowego do nauk społecznych i politycznych. Polityka miałaby się wówczas stać racjonalna.

W okresie przed rewolucją naukową, wychodzono z założenia, że natura jest racjonalna, bo Bóg, jej stwórca, jest racjonalny. Po rewolucji odkryto racjonalność natury w niej samej. Badanie świata naturalnego nie jest już poznawaniem Boga. Natura jest mechanizmem. Kepler pisał, że *caelestic machina* była nie *instar divini animalis*,

*sed instar horologii*, a Galileusz często wypowiadał się podobnie, szczególnie w swoim słynnym adagio *uniuersum horologium est*, wszechświat jest zegarem. Bóg jest inżynierem. Byłby złym inżynierem, a nie jest, gdyby nieustannie angażował się w działania tego mechanizmu. Ostatecznie staje się zbyt cenny. Pierre Simon de Laplace zaprezentował Napoleonowi *Systeme du Monde*. Ten zapytał go: „Napisałeś olbrzymią książkę o systemie świata bez jakiegokolwiek wzmianki o Stwórcy wszechświata?” Laplace odpowiedział: „Sir, nie potrzebowałem żadnej takiej hipotezy” (Je n’aurais pas besoin de cette hypothèse-là). Napoleon powiedział o tym Lagrange, który wykrzyknął: „Ah! c’est une belle hypothèse; ça explique beaucoup de choses”. [25].

## 5. Idea algorytmu

Idea algorytmu przenika wszelkie nauki, a piękno algorytmicznego ujęcia koreluje z łatwością rozumienia.

Różne algorytmy były używane jeszcze przed naszą erą. Babilończycy matematycy już ok. 2500 r. p.n.e., a Egipcjacy ok. 1500 r. p.n.e. obliczali iloraz algorytmicznie. Grecy matematycy korzystali z sita Eratostenesa dla znajdowania liczb pierwszych, a z algorytmu Euklidesa dla znajdowania największego wspólnego dzielnika. W IX w. Arabii używano algorytmów kryptograficznych dla deszyfracji.

Nazwa „algorytm” wywodzi się od nazwiska urodzonego na terenach obecnego Uzbekistanu matematyka perskiego Abu Abdullaha Muhammada ibn Musy al-Chwarizmiego, a raczej jego zlatynizowanej wersji „al-Chwarizmi” [55]. Łacińskie „algorithmus” powstało z kombinacji „algorism” i greckiego „arithmós” (liczba) [72]. „Algorithmus” (algorismus) oznaczało wykonywanie operacji arytmetycznych na liczbach zapisanych cyframi arabskimi w odróżnieniu od wykonywania tych operacji na liczbach zapisanych cyframi rzymskimi.

Robert z Chester, który był pierwszym tłumaczem na łacinę dziś już zaginionej książki al-Chwarizmi (1969, s. 411), swój przekład – odnaleziony w XIX w. – zaczyna słowami:

*Dixit Algoritmi: laudes deo rectori nostro atque defensori dicimus dignas.*

*Algoritmi powiedziało: niech pochwalony będzie Bóg, nasz Pan i Wspomożyciel.*

Około 1143 r. [77] dokonane zostało streszczenie tego dzieła, dzisiaj znane jako Kodeks z Salem (Salem Codex) [16]. Na początku czytamy:

*Incipit liber algorizmi: omnis sapientia sive scientia a domine Deo; sicut scriptum est: Hoc quod continent omnia scientiam habet, et iterum: Omnia in mensura et pondere et numero constituisti.*

*Wstęp do księgi algorytmu. Wszelka mądrość i wszelka wiedza pochodzi od Boga naszego Pana; jak jest napisane: [Mdr 1:7] Ten, który ogarnia wszystko ma pełnię wiedzy, i dalej: [Mdr 11:20] Wszystko urządził według miary i liczby, i wagi!*

Jak zauważa się [16], zastosowana forma gramatyczna dowodzi, że autor nie miał świadomości, że chodzi o nazwisko. W tym tekście nazwa „algorizmi” została po raz pierwszy – w zachowanym piśmiennictwie – użyta na oznaczenie procedury:

*Der Gebrauch des Nominativus algorizmus beweist, dass das Bewusstsein, dass Algorizmus der Name eines Mannes sei, bei dem Verfasser der Abhandlung schon verloren gegangen war. Er hielt offenbar dieses Wort für den Namen der Rechenkunst selbst.*

*Użycie nominatiwu algorizmus dowodzi, że świadomość, że Algorizmus było czymś imieniem, była już zagubiona przez autora tej rozprawki. W oczywisty sposób traktuje to słowo jako nazwę sztuki rachowania.*

Zastanawiające, czy autor Kodeksu z Salem odwołuje się do wszechogarniającej wiedzy Boga, aby było wzniośle, bo taki był obyczaj, czy ma jakieś przeczucie roli i miejsca algorytmów w dziele stworzenia. Jeśli to drugie, to można wskazywać go jako tego, kto antycypował podstawową ideę filozofii informatycznej, czyli że działaniem się świata kierują algorytmy.

W wierszu łacińskim napisanym dla potrzeb dydaktyki, a przypisywanym Aleksandrowi de Villa Dei (Alexander de Villedieu) *Carmen de Algorismo lub Algorismus metricus*<sup>5</sup> czytamy:

*Hinc incipit algorismus.  
Haec algorismus ars praesens dicitur in qua  
Talibus Indorum fruimur bis quinque figuris  
0 9 8 7 6 5 4 3 2 1,*

*Tu zaczyna się algorytm.  
Ta nowa sztuka jest nazwana algorytmem, w którym  
z tych dwukrotnie pięciu cyfr  
0 9 8 7 6 5 4 3 2 1,*

od Hindusów czerpiemy taką korzyść.

Algorytmy pozostają w związku ze sposobem kodowania informacji. Inaczej mówiąc, zmiana sposobu kodowania może wiązać się ze zmianą algorytmu. Może być tak, że ta zmiana jest radykalna – jak można przypuszczać – tak jak w przypadku algorytmów

---

<sup>5</sup> [https://upload.wikimedia.org/wikipedia/commons/3/35/Carmen\\_de\\_Algorismo.pdf](https://upload.wikimedia.org/wikipedia/commons/3/35/Carmen_de_Algorismo.pdf) [13.10.2020].



fizycznych przetwarzających kod biologiczny. Dodajmy za Marciszewskim [73], że przez algorytmy fizyczne rozumiemy algorytmy sterujące przetwarzaniem informacji, dokonującej się w rzeczywistości fizycznej – tzn. te, które przetwarzają informację składającą się na świat – w odróżnieniu od algorytmów symbolicznych, które piszemy i za pomocą których komputery przetwarzają informację przez nas zakodowaną. Algorytmy naturalne realizują obliczenia naturalne. Obliczenia poznawcze – te, których my dokonujemy, przeprowadzamy za pomocą algorytmów symbolicznych.

Algorytmy nie tylko powinny – co jest oczywiste – być poprawne, czyli dawać na wyjściu prawdziwy wynik, ale nadto jak mówi Knuth [55]:

*we want good algorithms in some loosely defined aesthetic sense. One criterion [...] is the length of time taken to perform the algorithm [...] Other criteria are adaptability of the algorithm to computers, its simplicity and elegance, etc.*

*chcemy dobrych algorytmów w pewnym luźnie określonym sensie estetycznym. Jednym z kryteriów [...] jest długość czasu, jaki zabiera wykonanie tego algorytmu [...] Innymi kryteriami są adaptowalność programu do komputerów, jego prostota i elegancja, etc.*

Chaitin [20] precyzuje pojęcie elegancji programu:

*a program is 'elegant,' by which I mean that it's the smallest possible program for producing the output that it does.*

*'elegantki' program to taki, który, tak rozumiem, jest najmniejszym możliwym programem dla wytworzenia danych wyjściowych.*

Jednocześnie dodaje, że:

*I'll show you can't prove that a program is 'elegant' – such a proof would solve the Halting problem.*

*Pokażę, że nie możemy dowieść elegancji programu – taki dowód byłby rozwiązaniem problemu stopu.*

Piękno algorytmów naturalnych, jak i ich dostępność dla ludzkiego umysłu jest dziedziczona przez algorytmy symboliczne.

Definicja algorytmu jest dziełem matematyków i logików XX w. Potrzeba takiej definicji ujawniła się w związku z programem Hilberta, który postulował tworzenie matematyki przez formalne przekształcenia symbolicznej reprezentacji wiedzy matematycznej. Te przekształcenia miały być takie, aby nie było sporu co do ich poprawnej realizacji. Ponadto, co jest jasne, miały prowadzić od prawdziwych zdań matematycznych do prawdziwych zdań matematycznych, czyli w ten sposób miała

być wykluczona ewentualna sprzeczność, jeśli wyjściowe dane nie były sprzeczne. Przynależność jakiegoś zdania do zbioru twierdzeń miała być rozstrzygana formalnie. Takie postawienie sprawy wymagało sprecyzowania pojęcia metody formalnej, która byłaby narzędziem realizacji takiego przedsięwzięcia. Wśród propozycji – które okazały się równoważne – szczególnie uznanie znalazła koncepcja opracowana przez Alana Turinga zwana dzisiaj maszyną Turinga. Algorytm to procedura, która jest wykonywalna za pomocą maszyny Turinga.

Choć koncepcja tak określonego algorytmu odniosła sukces, nie oznacza to, że zaprzestano – w tym również Turing [104] – rozważań nad modyfikacją pojęcia algorytmu.

Dodajmy, że (angielskie) słowo „computer” jeszcze w XIX w., a nawet w 1936 r. – kiedy Turing opublikował *On Computable Numbers, with an Application to the Entscheidungsproblem* (1936–37) – było używane na wskazanie urzędnika, który wykonywał uciążliwe numeryczne obliczenia [22]. Tak rozumiane „computer” oznaczałoby rachmistrza. Teksty po polsku, w których „computer” przekładane jest na „komputer” pozbawione są ewentualnych skojarzeń obecnych w tekstach anglojęzycznych. W szczególności kojarzenie „computer” z „rachmistrz” ma znaczenie dla rozumienia tekstów Turinga.

## 6. Rzeczywistość a informacja

Wszystko, co wiemy jest informacją<sup>6</sup>. To, co poznajemy jest informacją. Jak pisze Luciano Floridi [35]:

*Reality in itself is not a source but a resource for knowledge.*

*Sama rzeczywistość nie jest źródłem, lecz zasobem wiedzy.*

Nie poznajemy niczego, co nie jest informacją. Jak stwierdza Stephen Wolfram [115]:

*[M]atter is merely our way of representing to ourselves things that are in fact some pattern of information, but we can also say that matter is the primary thing and that information is our representation of that. It makes little difference, I don't think there's a big distinction – if one is right that there's an ultimate model for the representation of universe in terms of computation.*

---

<sup>6</sup> Stacewicz (2011, §1. Różne wymiary informacji) znakomicie omawia pojęcia informacji i jej związki z wiedzą.

*Materia jest jedynie naszym sposobem reprezentacji sobie rzeczy, które w istocie są wzorcami informacji, lecz możemy też powiedzieć, że materia jest rzeczą pierwotną a informacja jest naszą jej reprezentacją. To czyni małą różnicę, i nie myślę, że jest to duże rozróżnienie – jeśli ma się rację, że istnieje ostateczny model reprezentacji wszechświata w terminach komputacji.*

Pozyskiwana informacja musi być jakoś reprezentowana. Reprezentacja umożliwia jej przechowywanie, komunikowanie i przetwarzanie. Każda informacja może być kodowana zero-jedynkowo. Sposób reprezentacji podporządkowany jest celowi, jakiemu ma służyć. Jak ujmuje to John Wheeler [113]:

*every physical quantity, every it, derives its ultimate significance from bits, binary yes-or-no indications.*

*każda fizyczna wielkość, wszystko co jest, wywodzi swoje ostateczne znaczenie z bitów, binarnego tak-lub-nie.*

Ta idea wyrażona może być skrótowo: it from bit, gdzie „it” – to, co istnieje, a „bit” odnosi się do informacji.

Konrad Zuse [121], rozwijając koncepcję zdigitalizowanych relacji przestrzennych, ideę rozumienia wszechświata jako komputera, pojęciu informacji przypisuje istotną rolę:

*In current expanded usage, the term “compute” is identical with “information processing.” By analogy, the terms “computer” and “information-processing machine” may be taken as identical.*

*W bieżącym poszerzonym zastosowaniu, termin „obliczać” jest identyczny z „przetwarzać informację”. Przez analogię jako identyczne mogą być wzięte terminy „komputer” i „maszyna przetwarzająca-informację”.*

Zuse był pierwszym, który zasugerował, że stany fizyczne wszechświata są obliczane przez sam wszechświat. Wskazywał na automaty komórkowe. Koncepcję automatów komórkowych opracował John von Neumann w związku ze swoimi poszukiwaniami podobieństw między komputerami a centralnym systemem nerwowym [84, 108, 109, 110].

Informacja może być przetwarzana algorytmicznie. Arystoteles, tworząc sylogistykę konstruuje formalny system przetwarzania informacji. Ta idea rozwijana jest w logice formalnej. Zwykle wskazuje się na Gottfrieda Leibniza jako tego, który podkreślał i wiązał rozwój wiedzy z zastosowaniami rachunkowego przetwarzania informacji.

Jeśli myślenie jest rachunkiem, a świat stworzony jest z liczb, to do wszelkiej prawdy, do której możemy dojść, dojdziemy drogą rachunkową. Zatem [67]<sup>7</sup>:

*Quo facto, quando orientur controversiae, non magis disputatione opus erit inter duos philosophos, quam inter duos Computistas. Sufficiet enim calamos in manus sumere sedereque ad abacos, et sibi mutuo (accito si placet amico) dicere: c a l c u l e m u s.*

Gdyby spór powstał, dysputa między dwoma filozofami nie wymagałaby większego wysiłku niż między dwoma rachmistrzami. Wystarczyłoby bowiem, aby wzięli ołówki w swoje ręce, usiedli przy swoich tabliczkach i jeden drugiemu (z przyjacielem jako świadkiem, gdyby zechcieli) powiedzieli: P o l i c z m y.

Teza ontologiczna o świecie jako stworzonym przez 1 za pomocą 0 otworzyła nowe perspektywy dla połączenia koncepcji informacji z metafizyką. Zachwalając swoją arytmetykę binarną, Leibniz [69] twierdził:

*tamen ubi Arithmeticae meae Binariae excogitavi, antequam Fohianorum characterum in mentem venirent, pulcherrimam in ea latere judicavi imaginem creationis, seu originis rerum ex nihilo per potentiam summae Unitatis, seu Dei.*

jednak gdy wymyśliłem moją arytmetykę binarną, zanim zaznajomiłem się z symbolami Fohy, uznałem w nich najpiękniejszy obraz stworzenia, czyli pochodzenia rzeczy z niczego dzięki najwyższej mocy Jedności, czyli Boga.

Idea ta tak bardzo fascynowała Leibniza, że przekazywał ją ojcu Grimaldi, matematykowi na dworze cesarza Chin w nadziei, że za jej pomocą doprowadzi do nawrócenia cesarza a wraz z nim chrystianizacji całych Chin [65].

Rachowanie jest czynnością, w której maszyna może zastąpić człowieka. W 1685 r., omawiając wartość dla astronomów wymyślonej w 1673 r. przez siebie maszyny liczącej, sprawniejszej niż pascalina i wykonującej wszystkie podstawowe działania arytmetyczne, pisał Leibniz [68] (zob. Davis [24]), że:

*For it is unworthy of excellent men to lose hours like slaves in the labor of calculation which could safely be relegated to anyone else if the machine were used.*

Nie jest godne wspaniałego człowieka tracić godziny jak niewolnicy w pracy rachunkowej, która bez obaw może być przekazana komukolwiek, gdyby użyć maszyny.

---

<sup>7</sup> Podobne stwierdzenia znajdują się w innych tekstach cytowanego tomu, np. na stronach: 26, 64–65, 125.

Charles Babbage, kiedy wraz z kolegą przygotowywał tablice matematyczne, zauważając mnóstwo błędów, sfrustrowany miał wykrzyknąć [89]:

*I wish to God these calculations had been executed by steam!*

*Na Boga, chciałbym te rachunki powierzyć parze!*

Konrad Zuse w wywiadzie udzielonym Uta Merzbach w 1978 r. mówił, że kiedy przyszło mu wykonywać żmudne rachunki inżynierskie, myślał<sup>8</sup>:

*It's beneath a man. That should be accomplished with machines.*

*To nie jest dla człowieka. To powinno być wykonane przez maszyny.*

motywowała go do podjęcia prac nad skonstruowaniem komputera [22].

Ten pragmatyczny argument z powyższymi argumentami natury metafizycznej może inspirować informatykę i rozwój jej narzędzi w kierunku sztucznej inteligencji. Wszelka prawda ma bowiem reprezentację liczbową, a myślenie jest reprezentowane przez operacje liczbowe, a to wszystko może wykonać maszyna.

Idea mechanicznego pozyskiwania wiedzy, *ars combinatoria*, mająca dawne korzenie, a w Europie propagowana i rozwijana przez lullystów, czyli tych którzy nawiązywali do koncepcji Rajmundusa Lullusa [101, 102], musiała być w XVII w. popularna, jeśli znajdujemy również literackie odniesienia do niej. Jonathan Swift, Irlandczyk, dwadzieścia jeden lat młodszy od Leibniza, w 1726 r. w *Gulliver's Travels* [91] literacko obrazuje taki zamysł:

*The first professor I saw, was in a very large room, with forty pupils about him. After salutation, observing me to look earnestly upon a frame, which took up the greatest part of both the length and breadth of the room, he said, "Perhaps I might wonder to see him employed in a project for improving speculative knowledge, by practical and mechanical operations. But the world would soon be sensible of its usefulness; and he flattered himself, that a more noble, exalted thought never sprang in any other man's head. Every one knew how laborious the usual method is of attaining to arts and sciences; whereas, by his contrivance, the most ignorant person, at a reasonable charge, and with a little bodily labour, might write books in philosophy, poetry, politics, laws, mathematics, and theology, without the least assistance from genius or study."*

*Pierwszy profesor, którego ujrzałem, znajdował się w wielkim pokoju, otoczony przez czterdziestu uczniów. Po przywitaniu się, gdy spostrzegł, że bardzo*

---

<sup>8</sup> Konrad Zuse interviewed by Uta Merzbach in 1968 (Computer Oral History Collection, Archives Centre, National Museum of American History, Washington D.C.).

*uważnie oglądam wielką maszynę zabierającą większą część pokoju, zapytał, czy nie budzi we mnie zdziwienia, że trudni się udoskonaleniem wiedzy spekulatywnej za pomocą operacji mechanicznych. Pochlebia sobie, że świat uzna ważność jego wynalazku i że wznioślejsza myśl nigdy w głowie człowieka nie powstała. Wiadomo, jak trudno przychodzi każdemu człowiekowi nauczyć się kunsztów i umiejętności, lecz dzięki jego wynalazkowi człowiek najbardziej nawet niewykształcony potrafi niewielkim kosztem i po lekkim ćwiczeniu ciała pisać książki filozoficzne, poetyczne, rozprawy o polityce, teologii i matematyce bez najmniejszej pomocy naturalnych zdolności lub nauk.*

## 7. Pojęcie paradygmatu i jego implementacje

Termin „paradygmat” wywodzi się z greki: παράδειγμα (parádeigma), co przekłada się na „przykład”, „wzorzec”, „szablon” lub „model wyjaśnienia”, „wzrost światła”, „światopogląd”. Termin „paradygmat” spopularyzował Thomas Kuhn w książce *Struktura rewolucji naukowych* [61]. Jednak termin ten był już używany przez Platona w *Timajosie* na oznaczenie modelu, wzorca, którego użył Demiurg, tworząc kosmos.

W metodologii przez „paradygmat” rozumie się wzorzec rozumowania lub postępowania badawczego w (dojrzałej) dyscyplinie naukowej.

Paradygmat obejmuje filozoficzne i metodologiczne założenia powszechnie i trwale przyjmowane przez uprawiających naukę na jakimś jej etapie rozwoju. Wiedzę dzieli się na paradygmatyczną, czyli naukową, i preparadygmatyczną, czyli przednaukową.

Paradygmat jest wzorcem uprawiania nauki. Nowy wzorzec, nowy paradygmat, odrzuca jako (już) nienaukowe niektóre problemy starej nauki i nadaje nowy sens tym, które pozostają w nowej nauce. Ponadto, co jest istotne, rozwiązuje problemy, z którymi nauka w uprzedniej wersji paradygmatu sobie nie radziła i wyznacza nowe pytania.

Galileusz głosił – co doprowadziło do wyznaczenia różnego od arystotelesowskiego paradygmatu nauki – że księga natury jest napisana językiem matematyki, dlatego ten język jest właściwy do jej poznania i rozumienia. Matematyczne przyrodznawstwo uprawiane jest zgodnie z paradygmatem Galileusza.

Zauważmy, że w czasach Galileusza stan wiedzy matematycznej był daleki od tego, jaki jest współcześnie. Matematyka w czasach Galileusza jest różna od tej, którą stosuje dzisiejsza nauka. Rozwój matematyki był sprzężony z postępem przyrodznawstwa. Na przykład Newton dla potrzeb swojej „filozofii naturalnej” tworzy rachunek różniczkowy i całkowy.

Tworzenie nauki zgodnie z paradygmatem Galileusza zaowocowało nie tylko głębszym poznaniem świata przyrody, lecz również przyniosło owoce w postaci technologii, co spowodowało rozwój przemysłu, a także zmiany stosunków społecznych [73].

Filozofia informatyczna głosi, że księga rzeczywistości naturalnej zapisana jest językiem algorytmiki i ten język jest właściwym językiem wiedzy zarówno o zjawiskach przyrodniczych, jak i wszelkich innych dostępnych poznawczo człowiekowi

w porządku naturalnym. To podejście wyznacza nowy paradygmat. Nazywamy go „paradygmatem Turinga”.

Paradygmat Turinga nie stoi w sprzeczności z paradygmatem Galileusza, raczej go uściśla i modyfikuje. Jednak ma właściwe dla paradygmatu konsekwencje, unieważniając pewne problemy przede wszystkim w obszarze biologii, psychologii oraz socjologii i otwierając perspektywy badań, które – mówiąc swobodnie – nie były widoczne lub nie były widoczne z perspektywy paradygmatu Galileusza, jak np. problematyka umysłu, życia społecznego i gospodarczego [73].

Gaston Bachelard [7] w swojej epistemologii historycznej wprowadził pojęcia bariery epistemologicznej (obstacle épistémologique) i pęknięcia epistemologicznego (rupture épistémologique). Nauka nie rozwija się tylko kumulatywnie. Pęknięcie epistemologiczne – termin spopularyzował Louis Althusser – następuje, kiedy ma miejsce integracja starej teorii do nowego paradygmatu. Można się zgodzić, że tego rodzaju sytuacja realizowana jest przez zastąpienie paradygmatu Galileusza paradygmatem Turinga.

Paradygmat ewolucyjny Darwina zdaje się być nie do uzgodnienia z paradygmatem Galileusza, a komponuje się i wzajemnie uzupełnia z paradygmatem Turinga. Paradygmat Turinga ma w swoim zasięgu nie tylko przyrodznawstwo, ale wszystko to, co tradycyjnie nazywano filozofią naturalną. Daje możliwości kompleksowych badań samoorganizujących się systemów adaptacyjnych, bez względu na ich typ (fizyczne, biologiczne, społeczne) [28].

## 8. Świat tworzony przez algorytmy

Pojęcie algorytmu jest podstawowe dla paradygmatu Turinga. Nie znaczy to, że pojęcie to jest ostatecznie zdefiniowane i zamknięte na zmiany oraz modyfikacje. Podobnie jak matematyka w przypadku paradygmatu Galileusza, jest żywe i sprzężone z rozwojem badań. Napisze Marciszewski [73]:

*Intuicja intelektualna oraz pomysłowość uczonego są tym, dzięki czemu mogą powstawać nowe algorytmy, które na tyle wzmocnią system informatyczny, że problemy w poprzedniej fazie nierozstrzygalne staną się możliwe do rozstrzygnięcia w sposób algorytmiczny. W nowym systemie powstaną nowe problemy nierozstrzygalne, ale znów jest szansa na pokonanie trudności dzięki twórczej intuicji. Okazuje się więc, że proces poznawania świata matematycznego z udziałem maszyn nigdy nie zamknięty w sensie posiadania ostatecznych wyników, ale nigdy też nie jest zamknięty w sensie niemożności dalszego rozwoju. Możliwy jest rozwój w nieskończoność.*

Turing nie tylko podał definicję algorytmu, maszyny Turinga, lecz również wskazał nowe obszary dostosowania pojęcia algorytmu do potrzeb badawczych.

Turing – przynajmniej wśród tych, którzy mieli dorobek naukowy w zakresie algorytmiki – był pierwszym, który powziął ideę paradygmatu, określanego jako informatyczny. *Computing Machinery and Intelligence* (1950) może być wciąż źródłem inspiracji w tworzeniu i rozwijaniu paradygmatu informatycznego. Konrad Zuse, kończąc rozważania w *Computing Machinery and Intelligence* (1950, s. 64) zauważa niedogodności systematycznej metody rozwiązania i pisze:

*We may hope that machines will eventually compete with men in all purely intellectual fields. But which are the best ones to start with? Even this is a difficult decision. Many people think that a very abstract activity, like the playing of chess, would be best. It can also be maintained that it is best to provide the machine with the best sense organs that money can buy, and then teach it to understand and speak English. This process could follow the normal teaching of a child. Things would be pointed out and named, etc. Again I do not know what the right answer is, but I think both approaches should be tried.*

*Możemy mieć nadzieję, że maszyny ostatecznie będą współzawodniczyć z ludźmi na wszystkich czysto intelektualnych polach. Od czego jednak najlepiej zacząć? Nawet to jest trudną decyzją. Wielu sądzi, że najlepsze byłoby bardzo abstrakcyjne działanie, jak gra w szachy. Można także utrzymywać, najlepsze byłoby pozyskanie maszyny z najlepszymi organami sensorycznymi, jakie można kupić i uczenie, jak rozumieć i mówić po angielsku. Ten proces postępowałby jak zwykłe uczenie dziecka. Wskazywane byłyby rzeczy i nazywane, etc. Znowu nie wiem, jaka jest właściwa odpowiedź, lecz myślę, że trzeba spróbować obu podejść.*

W tej heroicznej fazie dziejów informatyki – jak określa to Marciszewski (2011, s. 165) – oprócz Turinga istotny wkład wnosi John von Neumann, kładąc podwaliny paradygmatu informatycznego. Von Neumann poszedł dalej – choć tym samym śladem co Turing – w postulowaniu rozumienia algorytmu. W niedokończonyj, pisanej przed śmiercią książce *The Computer and the Brain* (1958) bada algorytmy, których nośnikiem byłoby żywe białko.

Rozważania możliwości sztucznej inteligencji, którą mogłaby przejawiać maszyna zbudowana według reguł i zasad paradygmatu mechanicystycznego, prowadzą do konkluzji, że jako taka nie dorówna inteligencji tej, którą przejawiają organizmy żywe [100].

Do tej heroicznej fazy dziejów należy również Konrad Zuse (<http://www.konrad-zuse.de>). Był on pionierem informatyki, choć jego nazwisko jest mniej powszechnie znane. Zuse już w latach 40. zbudował pierwszy w pełni programowalny komputer Z3. Język programowania Plankalkül wyprzedzał to, do czego inni doszli później. Dodajmy, że skala i wartość osiągnięć technicznych Zuse jest dyskusyjna [22].

Zuse w *Rechnender Raum* [117] jako pierwszy mówi o wszechświecie jako swoistej sieci komputerowej. Nie ogłasza, że ma pełną teorię wszystkiego w postaci jakiegoś algorytmu liczącego wszechświat, lecz w tym tekście jako pierwszy jasno formułuje



taką ideę. Wyniki swoich dalszych przemyśleń opublikował w *Rechnender Raum* [117] oraz w *Nature as Computation* [121]. W *Der Computer* [120] wspomina:

*Es geschah bei dem Gedanken der Kausalität, dass mir plötzlich der Gedanke auftauchte, den Kosmos als eine gigantische Rechenmaschine aufzufassen. Ich dachte dabei an die Relaisrechner: Relaisrechner enthalten Relaisketten. Stößt man ein Relais an, so pflanzt sich dieser Impuls durch die ganze Kette fort. So müsste sich auch ein Lichtquant fortpflanzen, ging es mir durch den Kopf. Der Gedanke setzte sich fest; ich habe ihn im Laufe der Jahre zur Idee des „Rechnenden Raumes“ ausgebaut. Es sollte freilich dreißig Jahre dauern, ehe mir eine erste konkrete Formulierung der Idee gelang.*

*Kiedy myślałem o przyczynowości nagle ogarnęła mnie myśl, aby pojąć kosmos jako jakąś gigantyczną maszynę liczącą. Myślałem przy tym o kalkulatorze przekaźnikowym: kalkulatory przekaźnikowe zawierają łańcuchy przekaźnikowe. Wzbudzi się jeden przekaźnik, to ten impuls będzie przekazywany przez cały łańcuch. Tak musiałyby być przekazywane kwanty światła – przeszło mi przez głowę. Myśl ta opanowała mnie; w przeciągu lat rozbudowywałem ideę „Rechnenden Raumes” (przestrzeń obliczeniowa). Trwało to właściwie trzydzieści lat, zanim doszedłem do jakiegoś konkretnego sformułowania.*

Dopiero w trzecim tysiącleciu idea świata jako komputera zaczęła wzbudzać większe zainteresowanie. Zaczęły być publikowane m.in. w *Scientific American* i *Spektrum der Wissenschaft* teksty takie, jak: „Czy wszechświat jest Wielkim Komputerem?”, „Czy Wszechświat jest Komputerem?”.

Zuse, kończąc *Rechnender Raum* [117] wymienia paradygmatyczne różnice między mechaniką klasyczną, mechaniką kwantową a swoją koncepcją, co przedstawiono w tabeli I.1.

TABELA I.1. Paradygmatyczne różnice między mechaniką klasyczną, mechaniką kwantową a koncepcją *Rechnender Raum*

lp	Fizyka klasyczna	Fizyka kwantowa	Przestrzeń obliczeniowa
1.	mechanika punktu	mechanika falowa	teoria automatów, algebra przełączników
2.	korpuskuła	fala-korpuskuła	stan przełącznika, cząstka cyfrowa (Digitalteilchen)
3.	analogowe	hybrydowe	cyfrowe
4.	analiza	równania różniczkowe	równania różnicowe i algebra Boole'a
5.	wszystkie wielkości ciągłe	niektóre wielkości skwantowane	wszystkie wielkości mają tylko wartości dyskretne

lp	Fizyka klasyczna	Fizyka kwantowa	Przestrzeń obliczeniowa
6.	żadnych wartości granicznych	poza prędkością światła żadnych wartości granicznych	minimalne i maksymalne wartości wszystkich wielkości
7.	nieskończona dokładność	relacja nieokreśloności	ograniczona dokładność rachunków
8.	przyczynowość w obydwu kierunkach czasu	tylko statystyczna przyczynowość, rozwiązanie tylko uprawdopodobniające	przyczynowość tylko w przyszłość, możliwe wprowadzenie terminów uprawdopodobniających, ale niekonieczne
9.		klasyczna mechanika wprowadzona jako statystyczna	prawa uprawdopodobniające fizyki kwantowej wyjaśniane przez strukturę przestrzeni
10.		formuły bazowe	przełączniki bazowe

Choć koncepcja Konrada Zuse świata jako wielkiego komputera jest dyskusyjna [22], to wskazane różnice między paradygmatami są interesujące z punktu widzenia filozofii informatycznej.

Alan Turing nie ograniczał swojego myślenia do kwestii informatycznych. Szukał nie tylko wiedzy na temat umysłu. Jego badania obejmowały również świat przyrody. Nie bez racji może być zakwalifikowany jako filozof przyrody [50]. Przykładem dociekań według paradygmatu, który tu określamy jako paradygmat Turinga, są badania których wyniki zawarł w *The Chemical Basis of Morphogenesis* [105].

Według mechanicyzmu wszystko, co jest i dzieje się w przyrodzie może być wyjaśniane za pomocą pojęć i praw mechaniki, ewentualnie mechaniki kwantowej. Według filozofii informatycznej wszystko, co jest przedmiotem poznania naukowego może być wyjaśnione jako algorytmiczne przetwarzanie informacji, analogicznie do działania maszyny Turinga i jej modyfikacji oraz uogólnień, czyli za pomocą pojęć i praw algorytmiki. Obliczenia we wszechświecie, obliczenia naturalne, wykonywane byłyby na wielu różnych poziomach organizacji: kwantowym, biologicznym, przestrzennym itd. Niektóre obliczenia byłyby dyskretne, niektóre ciągle [70].

Różnicę między jednym a drugim, między paradygmatem Galileusza a paradygmatem Turinga, można wskazać obrazowo: w koncepcji nauki w paradygmacie Galileusza świat jest dziełem Inżyniera mechanika, a w koncepcji nauki w paradygmacie Turinga świat jest dziełem Programisty. Jeśli z paradygmatem Galileusza można wiązać *Deus ex machina*, to z paradygmatem Turinga wiązałibyśmy zwrot: *Deus ex AI*.

Celnie, z uwzględnieniem kontekstu historycznego, paradygmat Turinga można scharakteryzować słowami Marciszewskiego [73], który w miejsce stwierdzenia Leibniza: *cum Deus calculat et cogitationem exercet, fit mundus*, stawia: *cum mundus calculat, fit mundus*, gdy świat rachuje, staje się świat. A może, zachowując analogię – mając na uwadze tłumaczenie „*cum Deus calculat et cogitationem exercet, fit mundus*” jako „gdy Bóg rachuje i wciela swe myśli w czyny, powstaje świat” – należy powiedzieć: *cum mundus calculat et algorithmum exercet, fit mundus?* „*Cum mundus calculat et algorithmum exercet, fit mundus*” tłumacząc, jako „gdy świat rachuje i wykonuje

algorytmy, staje się świat”. Świat Galileusza został obliczony, a świat Turinga na podstawie aktualnego stanu sam rachuje swój stan przyszyły [18].

W dyskusji Marciszewski pyta, czy zwroty „*calculat*” i „*algorithmum exercet*” są w paradygmacie Turinga równoznaczne, a przynajmniej równozakresowe. Jeśli tak, to stosunek między nimi trzeba by oddać nie przez „*et*”, ale (np.) „*id est*”. I rozważa, o co chodzi Leibnizowi, gdy po „*calculat*” dodaje „*et cogitationes exercet*”. Czy nie uważał on, że myślenie Boga utożsamia się z rachowaniem? To po co byłby ten dodatek? Dodajmy, że ten ostatni postawiony przez Marciszewskiego problem jest tematem wielu rozważań, a Louis Couturat [23] jako motto *La logique de Leibniz*, swojego fundamentalnego tekstu o logice Leibniza, wybrał właśnie skróconą postać „*cum Deus calculat... fit mundus*”. Gdyby pozostać przy skróconej wersji myśli Leibniza, to „*cum mundus algorithmum exercet, fit mundus*” wprost wyrażałoby ideę filozofii informatycznej.

W świecie Galileusza trwa wieczny ruch określony przez prawa fizyki. Sam świat pozostaje wieczny i niezmienny. Świat nie jest jednak wieczny: ma początek i będzie miał koniec. Nie jest niezmienny: ewoluje. Darwin wykazał ewolucję świata ożywionego. Współczesna fizyka stwierdza historyczność, ewoluowanie świata materialnego. Historia uczy o ewolucji świata społecznego.

Prawa mechaniki mówią o poruszaniu się kół, trybów i innych części maszyny, nie mówią o tym, że sama maszyna się zmienia. Ona po prostu jest zanurzona w świecie przestrzenno-czasowym. Prawa algorytmiki mówią o przetwarzaniu nie tylko części, składowych świata, lecz również (całego) świata.

## 9. Paradygmat algorytmiczny w nauce

Rozważmy przykładowe kwestie, których ujęcie w paradygmacie Turinga jest różne od ujęcia w paradygmacie Galileusza.

### 9.1. Czy paradygmat Turinga jest owocny?

Czy zmiana języka matematyki na język algorytmiki prowadzi do nowych pytań i umożliwia znajdowanie odpowiedzi na pytania, na które nie znajdujemy odpowiedzi w paradygmacie Galileusza?

W przypadku paradygmatu Galileusza naturalne jest pytanie, kto oblicza. W przypadku paradygmatu Turinga na pytanie, co przetwarza informację składającą się na świat, odpowiedź jest prosta: świat. Algorytm jest składową świata tak, jak dane i programy są składową komputera i jak jedno i drugie, dane i programy, są tak samo kodowane. Powtórzmy wyrażając to sentencję:

*cum mundus calculat et algorithmum exercet, fit mundus.*

Owocność poznawcza paradygmatu Turinga może się też – co brzmi paradoksalnie – przejawiać w stwierdzeniu, że niektóre procesy przyrodnicze i umysłowe nie są obliczalne. Sam Turing, mając na uwadze istnienie nieobliczalnych liczb rzeczywistych, wskazywał możliwość, że fizyka mózgu może nie być obliczalna oraz dopuszczał możliwość nieobliczalnych systemów fizycznych (Copeland et al., 2016, The physical computability thesis).

Owocność praktyczna paradygmatu informatycznego przejawia się w zastępowaniu technologii mechanicznych przez technologie informatyczne. To jest naoczne. Dokonuje się postęp cywilizacji, o jakim mówił Alfred Whitehead [114]:

*Civilization advances by extending the number of important operations which can be performed without thinking about them. Operations of thought are like cavalry charges in a battle – they are strictly limited in number, they require fresh horses, and must only be made at decisive moments.*

*Cywilizacja wzrasta przez powiększanie ilości znaczących operacji, których wykonanie nie wymaga myślenia o nich. Operacje myślowe są jak szarżę kawalerii w bitwie – są ściśle ograniczone co do ilości, wymagają rześkich koni i muszą być wykonane w decydujących momentach.*

Tak rozumiana cywilizacja zrealizuje się poprzez rozwój sztucznej inteligencji, która staje się „szarżą kawaleryjską” współczesnego świata.

Stanisław Krajewski [57] z tym, co tutaj nazywamy paradygmatem Turinga wiąże wielorakie nadzieje poznawcze:

*It seems to me that philosophy has entered a new condition because of the advent of computers. [...] I wish to point out something more fundamental – a new kind of experience with which we have familiarized ourselves because of computers. Much more has happened than the obvious, though still remarkable, ‘shrinking’ of the globe due to the ease of communication with nearly every spot on earth; even more amazing is the fact that so much can be recreated or simulated by programming. The philosophy of mind has been deeply affected by this: indeed, a cognitive science has arisen that conceives the mind as a biological computer. To understand it, the knowledge of logic should be useful. After all, logic, which emerged as the result of an analysis of thinking and thought patterns, was used to build computers, and computers, in turn, according to their enthusiasts, are about to acquire the ability to think. If so, then, however “artificial” this thinking could be, it would amount to not only information processing but to understanding as well.*

*Wydaje mi się, że filozofia znalazła się w nowej sytuacji z powodu pojawienia się komputerów. [...] Chcę wskazać coś bardziej fundamentalnego – nowy rodzaj doświadczenia, z którym zapoznaliśmy się dzięki komputerom. Znacznie więcej*

zdarzyło się niż oczywiste, chociaż wciąż godne uwagi, 'skurczenie się' globu, z powodu łatwości komunikacji z prawie każdym punktem ziemi; a nawet bardziej zadziwiający jest fakt, że tak wiele może być odtworzone lub symulowane programistycznie. Filozofia umysłu została tym głęboko dotknięta: rzeczywiście, powstała kognitywistyka, która pojmuje umysł jako biologiczny komputer. Dla zrozumienia tego powinna być użyteczna wiedza logiczna. Przede wszystkim, logika, która wyłoniła się jako wynik analizy myślenia i wzorców myśli, została użyta do budowy komputerów, a komputery, z kolei, według ich entuzjastów, są blisko od nabywania umiejętności myślenia. Jeśli tak, to, jakkolwiek „sztuczne” mogłoby to być myślenie, byłoby czymś równym nie tylko przetwarzaniu informacji, lecz również jej rozumieniu.

## 9.2. Poznanie umysłu

Paradygmat Turinga jest właściwy i owocny w badaniu umysłu. W tym obszarze wiedzy paradygmat Turinga odnosi największe sukcesy, tak że nawet można odnieść wrażenie, że jest to jego zasadniczy obszar zastosowania.

Zagadnienie umysłu w perspektywie paradygmatu informatycznego podjął Turing w związku ze śmiercią w 1930 r. swego przyjaciela z czasów szkolnych Christophera Morcoma. W 1932 r. będąc w odwiedzinach w domu rodzinnym Morcoma wyraził przekonanie, inspirowane lekturą książki Arthura S. Eddingtona *The Nature of The Physical World* [31], że mózg nie działa deterministycznie, a wolna wola ma oparcie w prawach fizyki kwantowej. Wynikiem jego przemyśleń jest również test, zwany dziś testem Turinga, który dał asumpt do algorytmicznego rozumienia umysłu i świadomości. Test ten stał się wzorcem dla innych, którzy stawiali sobie za cel pełną identyfikację umysłu [57]. Nowa multidyscyplinarna nauka, kognitywistyka, stała się polem szerokiej współpracy badaczy różnych aspektów umysłu i mózgu.

Osiągnięcia wiedzy o umyśle są znaczące. Dostarczają argumentów na rzecz negatywnej odpowiedzi na tytułowe pytanie postawione przez Wodzisława Duchą<sup>9</sup>: "Why Minds cannot be Received, but are Created by Brains". Życie po śmierci ma być mitem [74]. Pyta prof. Duch: „Czy syn człowieczy znajdzie wiarę [...] w cywilizacji informatycznej?”. Podejmuje temat w *Katolicyzm po kognitywistyce: O nową teologię umysłu*<sup>10</sup>.

Rozważmy założenia teologiczne tej dyskusji o relacji duch-ciało. Czy teologia rzeczywiście głosi to, co zakłada się w tej dyskusji? Zauważmy, że założenie rozdzielności duszy i ciała nie jest konieczną tezą teologii katolickiej. Nie przyjmuje tego jej nurt tomistyczny. Napisze Bocheński [10]:

*Tym bardziej stanowczo wypada powiedzieć, że mniemanie, jakoby człowiek składał się z dwóch kawałków, ciała i duszy, jest bardzo nędznym zabobonem. Cała*

<sup>9</sup> <https://repozytorium.umk.pl/bitstream/handle/item/5064/SetF.2017.014%2CDuch.pdf> [05.09.2019].

<sup>10</sup> [http://teologia.deon.pl/katolicki-obraz-natury-ludzkiej-i-nauki-kognitywne/\[05.09.2019\].](http://teologia.deon.pl/katolicki-obraz-natury-ludzkiej-i-nauki-kognitywne/[05.09.2019].)

*nasza nauka i wszyscy poważni myśliciele odrzucają go stanowczo. Aby tylko jeden przykład podać, św. Tomasz z Akwinu, jeden z największych myślicieli chrześcijaństwa, przeczy stanowczo, by dusza ludzka była „substancją zupełną”, tj. kawałkiem i broni poglądu, że jest „treścią (forma) ciała”.*

Czy tomistyczne ujęcie relacji duszy i ciała wpisuje się w paradygmat informatyczny? Na te i inne pytania nie zamierzamy tu dawać odpowiedzi, zauważmy jednak, że chrześcijanie głoszą zmartwychwstanie z duszą i ciałem, że koniec tego świata nie jest końcem świata w ogóle. Jak czytamy w Apokalipsie (21:1):

*I ujrzałem niebo nowe i ziemię nową,  
bo pierwsze niebo i pierwsza ziemia przeminęły,  
i morza już nie ma.*

Koniec świata byłby wówczas, gdyby wszystkie (fizyczne) algorytmy przestały działać w kierunku doskonalenia, bo świat osiągnąłby stan doskonały.

Filozofia materialistyczna przyjmuje koncepcję umysłu jako obiektu wyłącznie materialnego. Mózg Lenina, który zmarł w 1924 r. został wypreparowany i poddany badaniom w dedykowanym do tego instytucie. Dążono do pozyskania wiedzy biologicznej o mózgu geniusza, a także, zachowując ciało Lenina dopuszczano możliwość jego ożywienia. To podejście dokonywało się w paradygmacie Galileusza. Taka biologiczna koncepcja badania zasadniczo zawęża metody w stosunku do paradygmatu Turinga.

Paradygmat Turinga otwiera naukę na spekulacje na temat umysłu, jakie czynione są w związku z twierdzeniem Gödla i jego wersjami, prowadzącymi do odrzucenia mechanistycznej koncepcji umysłu [58].

### 9.3. Przewidywanie

Naukę uprawiamy m.in., aby móc przewidywać. Jak to sformułował filozof pozytywizmu August Comte:

*Savoir pour prévoir, prévoir pour pouvoir.*

*wiedzieć, aby przewidzieć; przewidzieć, aby móc działać.*

W paradygmacie mechanicystycznym niedostateczność przewidywania wyjaśniamy niedostatkiem relevantnych danych lub – ewentualnym – brakiem dostatecznej wiedzy na temat praw rządzących branżą pod uwagę rzeczywistością.

Paradygmat mechanicystyczny odnosi sukcesy w obszarze zjawisk makropryrocznych: potrafimy z dokładnością ograniczoną tylko błędami przyrządów obserwacyjnych przewidywać ruchy ciał niebieskich. Trochę gorzej wypada to na poziomie

mikro, ale działa. Kiedy jednak przewiduje się zgodnie z tym paradygmatem zjawiska społeczne, np. gospodarcze, to nawet upraszczając gospodarowanie – jak to czyniono w gospodarce centralnie planowanej – poprzez administrowanie cenami, wielkością produkcji, zasadami dystrybucji i innymi elementami, mającymi wpływ na wynik gospodarczy, doświadcza się braku przewidywalności. Dlaczego? Może po prostu dlatego, że założono mechanicystyczny model funkcjonowania gospodarki.

Aby poznać działanie gospodarki musimy stworzyć algorytmy symboliczne, które będą zgodne, czyli pozwolą na obliczanie wyników takich, jakie są wynikiem działania algorytmów rzeczywistych, czyli tych, według których obliczany jest przyszły stan gospodarki na podstawie stanów przeszłych. Jeśli uda się nam stworzyć trafne modele algorytmiczne, choćby tylko niektórych procesów gospodarczych, to niekoniecznie uda się nam przewidzieć wyniki algorytmów rzeczywistych. Mogą być przynajmniej trzy tego powody:

1. Algorytm symboliczny źle symuluje algorytm życia gospodarczego.
2. Algorytm symboliczny działa wolniej niż algorytm rzeczywisty, którego jest modelem.
3. Zawodzi system przekazu danych, na których operuje algorytm symboliczny.

Świat już dziś opleciony jest siecią informatyczną, a choć jej rozwój rodzi obawy o możliwość zachowania prywatności, a przede wszystkim wolności, w szczególności od manipulacji, nic nie wskazuje na zahamowanie tego. Dzięki pozyskiwaniu aktualnych danych meteorologicznych możliwe staje się coraz lepsze przewidywanie pogody. Czy życie gospodarcze jest bardziej „kapryśne” niż pogoda, czy też wciąż nie mamy dostępu do wystarczających zasobów danych lub nie mamy algorytmów symbolicznych dobrze symulujących algorytmy życia gospodarczego?

Rozważyć trzeba również to, że przewidywania gospodarcze mają wpływ na zachowania ludzi i najzwyczajej w świecie dochodzi do samounicestwienia tych przewidywań. Jak na razie najlepiej w gospodarce radzą sobie ci, którzy mają intuicję gospodarowania i mają dostęp do relewantnych danych.

## 9.4. Ruch maszyny a ewolucja algorytmiczna

Awaria, „śmierć” maszyny jest wadą, której powodem może być niedoskonałość konstrukcji, wadliwość zastosowanych materiałów lub błędy wykonania. Gdyby człowiek, w ogóle świat natury, był dziełem inżyniera mechanika, to śmierć wskazywałaby na brak kompetencji inżynierskich.

Ujmując rzecz naturalistycznie, przyroda stworzyła wyrafinowane konstrukcje, takie jak organizmy, materię ożywioną. O poziomie wyrafinowania świadczy to, że człowiekowi wciąż nie udało się stworzyć jakiegokolwiek formy materii ożywionej, a także wiedza na temat życia wciąż jest płytka. Organizmy są śmiertelne, zresztą wbrew oczekiwaniom tych organizmów. Co ograniczało przyrodę, aby wytworzyć osobniki, które są wieczne? Ujmując rzecz z perspektywy mechanicystycznej

pytamy się: co stało na przeszkodzie, aby wytworzony był mechanizm samonaprawy i odmładzania? Akurat w tym zakresie człowiek, nauka, ma pewne osiągnięcia.

Powyższa kwestia inaczej jednak wygląda, kiedy ujmie się ją z perspektywy paradygmatu algorytmicznego. Dobry algorytm to taki, który zatrzymuje się po skończonej ilości wykonanych instrukcji. Jeśli życie jest realizacją pewnego algorytmu, to koniec działania tego algorytmu wskazuje na kompetencje programisty: algorytm przestał działać, kiedy osiągnął wynik. Wszechświat jako dobry algorytm ma początek i będzie miał koniec, kiedy wykona zadanie, dla którego został „napisany”.

Śmierci, końca działania w świecie przyrody Galileusza, nie sposób opisać, nie zakładając jakiegokolwiek wady, jakiegoś zużywania się, wyczerpania się. Dla algorytmów koniec ich działania jest nie tylko naturalny, ale i jest ich oczekiwaną własnością. Z tej perspektywy śmierć jawi się jako spełnienie, wypełnienie. Organizmy żywe, realizowane przez algorytmy ewolucyjne, giną, aby dać miejsce doskonalszym. Te, które osiągnęłyby doskonałość mogłyby trwać wiecznie.

Jak celem ludzkiego działania jest dobro, co stwierdzał Platon [Gorgias 499e], a Arystoteles [3] powie:

*Wszelka sztuka i wszelkie badanie, a podobnie też wszelkie zarówno działanie jak i postanowienie, zdają się zdążać do jakiegoś dobra i dlatego trafnie określono dobro jako cel wszelkiego dążenia,*

tak też dobro byłoby celem działania algorytmów.

W paradygmacie Galileusza, jak to jest w fizyce newtonowskiej, czas i przestrzeń są bezgranicznym „naczyniem”, w którym przebiegają procesy fizyczne. W przypadku paradygmatu Turinga czas i przestrzeń są właściwościami algorytmów. Są im immanentne. Kiedy algorytm wypełni się, osiągnie doskonałość, czas – dla niego – przestaje płynąć, przynajmniej ten, który był wyznaczany przez rytm wykonywania algorytmu.

Czy idąc tym tropem myślenia, że ewolucja prowadzi do doskonalenia – skonstruowanie przez człowieka komputera doskonalszego od człowieka, stworzenie superinteligencji, jakichś robotów Čapka – doprowadzi do sytuacji, w której algorytm życia człowieka zakończy działanie, bo człowiek wypełnił już swoje zadanie? [12], a może – jak przewiduje to Kurzweil [62]:

*The Singularity will allow us to transcend these limitations of our biological bodies and brains. We will gain power over our fates. Our mortality will be in our own hands. We will be able to live as long as we want (a subtly different statement from saying we will live forever). We will fully understand human thinking and will vastly extend and expand its reach. By the end of this century, the nonbiological portion of our intelligence will be trillions of trillions of times more powerful than unaided human intelligence.*



*Osobliwość pozwoli nam przekroczyć ograniczenia naszych biologicznych ciał i mózgow. Uzyskamy władzę nad naszymi losami. Nasza moralność będzie w naszych własnych rękach. Będziemy zdolni żyć tak długo, jak zechcemy (subtelnie różne stwierdzenie od powiedzenia, że będziemy żyli wiecznie). Będziemy w pełni rozumieć ludzkie myślenie i będziemy szeroko poszerzać i poszerzać jego zasięg. Pod koniec tego wieku, nie-biologiczna część naszej inteligencji będzie tryliony trylionów razy potężniejsza niż niewspomagana ludzka inteligencja.*

## 9.5. Rozwój nauki

Nauka jest przedsięwzięciem historycznym. Obrazowo określa to Marciszewski [73]:

*Nauka nowożytna to niezmierny dziś ocean wiedzy, a myśl i dorobek Galileusza, w połączeniu z pionierskim dziełem Kopernika, jest jak ujście doń rzeki toczącej swe wody wcześniej przez dwa tysiąclecia. Jeśli zapytać, skąd ten nurt wypływa, gdzie i jakie są jego źródła, to nasza rzeczna metafora nadal się sprawdza. Okazuje się bowiem, że jest to tak, jak w przyrodzie. Dające się rozpoznać źródło stanowi uchwytny tej rzeki początek, ale on z kolei ma swe początki w sączących się niewidzialnie, pochowanych w murawie strumyczkach, bez których by nie zaistniało nasze oznaczone na mapie źródło.*

Osiągnięcia nauki zawdzięczamy poprzednikom. Głosił Jan z Salisbury, powtarzając za znanym z próby uzgodnienia filozofii Platona z filozofią Arystotelesa, Bernardem z Chartres [32] (Saresberiensis, 1159, III. CAP IV):

*nos esse quasi nanos, gigantium humeris incidentes, ut possimus plura eis et remotiora videre, non utique proprii visus acumine, aut eminentia corporis, sed quia in altum subvehimur et extollimur magnitudine gigantea.*

*Jesteśmy podobnie jak karły siedzące na barkach gigantów. W zasięgu wzroku mamy więc więcej rzeczy i widzimy dalej niż oni. Nie jest tak ani dlatego, że mamy ostrzejszy wzrok, ani dlatego, że jesteśmy więksi; lecz dlatego, że jesteśmy niesieni i wyniesieni przez wielkość gigantów.*

Newton, którego *Philosophiæ Naturalis Principia Mathematica* [80] otwiera erę nowoczesnej nauki, napisze do Roberta Hooke'a (1675):

*If I have seen further, it is by standing on the shoulders of giants,*

*zobaczyłem dalej (zrozumiałem więcej), bo stanąłem na barkach gigantów.*

Żadne pokolenie nie rozwiązało i – jak to uzasadnia filozofia informatyczna – nie rozwiąże wszystkich problemów, pozostawiając je przyszłym pokoleniom. Przeczuwał to już Newton [112]:

*To explain all nature is too difficult a task for any one man or even for any one age. 'Tis much better to do a little with certainty, & leave the rest for others that come after you, than to explain all things by conjecture without making sure of any thing.*

*Wyjaśnienie wszystkiego w przyrodzie jest zbyt trudnym zadaniem dla każdego jednego człowieka lub nawet dla jednego pokolenia. 'Znacznie lepiej jest zrobić niewiele, ale porządnie, & pozostawić resztę dla innych, którzy przyjdą po tobie, niż wyjaśnić wszystkie sprawy przez przypuszczenia bez pewności jakiejś rzeczy.*

Newton sam o sobie powie Brewsterowi [13]:

*I do not know what I appear to the world; but to myself I seem to have been only like a boy playing on a seashore, and diverting myself in now and then finding a smoother pebble or a prettier shell than ordinary, whilst the great ocean of truth lay undiscovered before me.*

*Nie wiem, jak mnie widzi świat; lecz mnie samemu zdaje się, że jestem jak mały chłopiec bawiący się na brzegu morza, zabawiający się znajdowaniem tu i tam wygładzonych kamyczków lub ładniejszych niż zwykle muszli, a przede mną rozciąga się wielki ocean nieodkrytych prawd.*

Z tworzeniem gmachu nauki jest podobnie jak z budową średniowiecznych katedr. Każdy, kto uczestniczył w budowie katedry miał różne cele prywatne i wносił różny wkład w jej powstanie, nie mając ani pewności, czy katedra ostatecznie będzie ukończona, ani jak będzie ostatecznie wyglądała. Nikt nie miał pewności, kiedy budowa się zakończy.

Newton, na którego popiersiu w Trinity College czytamy: *Qui genus humanum ingenio superavit* (nie ma większego intelektu wśród ludzi) – który sformułował mechanikę (newtonowską), wydającą się ostateczną teorią fizykalną – nie żywił przekonania o możliwości wyczerpania przez naukę wiedzy o świecie. Dzisiaj dzięki filozofii informatycznej wiemy, że jego przecucie było zasadne. Nauka ery informatycznej, o czym pisze Marciszewski [73], będzie w stanie niekończącego się rozwoju, nie wyczerpując wszystkich konsekwencji odkrytych prawd.

Kolejne pokolenia badaczy będą zasoby wiedzy powiększać, korygować, zgłębiać, ale i tak pozostaną obszary, o których będzie można powtórzyć za Emilem du Bois-Reymondem, niemieckim fizjologiem, przekonanie *ignoramus et ignorabimus* (nie wiemy i wiedzieć nie będziemy) wypowiedziane w Lipsku na wykładzie *Über die Grenzen des Naturerkennens* [29] (O granicach poznania przyrody) w Gesellschaft

Deutscher Naturforscher und Ärzte. Mówił on, że wobec zagadek świata materialnego badacz przyrody jest od dawna przywykły do ludzkiej niechęci wypowiedzenia swojego 'Ignoramus' (nie wiemy). Spojrzenie na przeszłe sukcesy prowadzi go do niezmaconej świadomości, że czego jeszcze nie wie, przynajmniej warunkowo mógłby wiedzieć, a kiedyś być może będzie wiedział. Wobec zagadki, czym są materia i siła, i jak są one do pomyślenia, musi za każdym razem zdecydować się na bardziej trudną prawdę: 'Ignorabimus' (nie będziemy wiedzieć):

*Gegenüber den Rätseln der Körperwelt ist der Naturforscher längst gewöhnt, mit männlicher Entsagung sein 'Ignoramus' auszusprechen. Im Rückblick auf die durchlaufene siegreiche Bahn trägt ihn dabei das stille Bewußtsein, daß, wo er jetzt nicht weiß, er wenigstens unter Umständen wissen könnte, und dereinst vielleicht wissen wird. Gegenüber dem Rätsel aber, was Materie und Kraft seien, und wie sie zu denken vermögen, muß er ein für allemal zu dem viel schwerer abzugebenden Wahrspruch sich entschließen: 'Ignorabimus'.*

Z przekonaniem du Bois-Reymonda, przynajmniej w matematyce, nie godził się David Hilbert [47]. Na kongresie matematyków w Paryżu w 1900 r. głosił, że wewnętrzny głos mówi:

*Da ist das Problem, suche die Lösung. Du kannst sie durch reines Denken finden; denn in der Mathematik gibt es kein Ignorabimus!*

*Jest problem, szukaj rozwiązania. Znaleźć możesz je za pomocą czystego myślenia; albowiem w matematyce nie istnieje Ignorabimus!*

W zakończeniu mowy pożegnalnej w Królewcu 8 września 1930 r. na spotkaniu Gesellschaft Deutscher Naturforscher und Ärzte twierdził, że [48]:

*Wir müssen wissen. Wir werden wissen.*

*Musimy wiedzieć. Będziemy wiedzieć*<sup>11</sup>.

Przekonanie to było znaczące dla rozwoju jego działalności badawczej. Inskrypcja tej treści znajduje się na nagrobku Hilberta na cmentarzu w Göttingen.

Podjęta przez Hilberta próba odrzucenia *Ignorabimus!* zaowocowała powstaniem informatyki i uzasadnieniem – paradoksalnie – odrzucenia przekonania Hilberta o możliwościach poznawczych metod formalnych.

---

<sup>11</sup> <http://math.sfsu.edu/smith/Documents/HilbertRadio/HilbertRadio.pdf> [12.07.2020].

## 10. Zakończenie

Kilka uwag i tez, nie do końca rozwiniętych i nie w pełni uzasadnionych pokazuje, że paradygmat Turinga przekracza granice nauk formalnych. Otwiera nowe perspektywy badań w naukach pozytywnych, a w szczególności zdaje się być jedynym właściwym paradygmatem poznania umysłu. Daje też okazję do spekulacji filozoficznych na temat świata jako zbudowanego z algorytmów.

### Literatura

- [1] Ares, J., Lara, J. A., Lizcano, D., & Martínez, M. A. (2018). Who Discovered the Binary system and Arithmetic? Did Leibniz plagiarize Caramuel? *Science and Engineering Ethics*, 24, 173–188. doi: 10.1007/s11948-017-9890-6.
- [2] Arystoteles. (1980). *O niebie*. Warszawa: PWN. (Przełożył, wstępem, komentarzem i skrowidzem opatrzył Paweł Siwek).
- [3] Arystoteles. (1982). *Etyka nikomachejska*. Warszawa: PWN. (Przełożyła Daniela Gromska).
- [4] Arystoteles. (1983). *Metafizyka*. Warszawa: PWN. (Przełożył, wstępem, komentarzem i skrowidzem opatrzył Kazimierz Leśniak).
- [5] Babbage, C. (1864). *Passages from the Life of a Philosopher*. London: Longman, Green, Longman, Roberts, & Green. [http://djm.cc/library/Passages\\_Life\\_of\\_a\\_Philosopher\\_Babbage\\_edited.pdf](http://djm.cc/library/Passages_Life_of_a_Philosopher_Babbage_edited.pdf).
- [6] Babbage, C. (2008). *Passages from the Life of a Philosopher*. Rough Draft Printing. <http://www.fourmilab.ch/babbage/lpae.html>.
- [7] Bachelard, G. (2002). *Kształtowanie się umysłu naukowego. Przyczynek do psychoanalizy wiedzy obiektywnej*. Gdańsk: Wydawnictwo słowo/obraz terytoria.
- [8] Bacon, R. (2010). *Mathematical Science*. In J. H. Bridges (Ed.), *The opus majus of Roger Bacon* (Vol. 1, pp. 97–404). Cambridge: Cambridge University Press. doi: 10.1017/CBO9780511709661.006.
- [9] Barrow, J. D. (2000). *The Book of Nothing. Vacuums, Voids and the Latest Ideas about the Origins of the Universe*. London: Vintage Books. <https://archive.org/details/indextocriticism00glor>.
- [10] Bocheński, J. M. (1994). *Sto zabobonów. Krótki filozoficzny słownik zabobonów* (2<sup>nd</sup> ed.). Kraków: Wydawnictwo PHILED spółka z o.o.
- [11] Boetii, A. M. T. S. (1867). *De institutione arithmetica* (Vols. 1–2; G. Friedlein, Ed.). Lipsiae: B. G. Teubneri. [https://la.wikisource.org/wiki/De\\_Arithmetica](https://la.wikisource.org/wiki/De_Arithmetica), [https://www.documenta-catholicaomnia.eu/03d/0480-0524,\\_Boethius,\\_De\\_Arithmetica,\\_LT.pdf](https://www.documenta-catholicaomnia.eu/03d/0480-0524,_Boethius,_De_Arithmetica,_LT.pdf).
- [12] Boström, N. (2014). *Superintelligence: Paths, Dangers, Strategies*. Oxford: Oxford University Press.
- [13] Brewster, D. K. H. (Ed.). (1855). *Memoirs of the Life, Writings, and Discoveries of Sir Isaac Newton* (Vol. 2). Edinburgh, London: Thomas Constable and Co. Hamilton, Adams, and Co. <https://archive.org/details/memoirslifewrit02brewgoog/page/n5/mode/2up>.
- [14] Burks, A. W., Goldstine, H. H., & von Neuman, J. (1987). Preliminary Discussion of the Logical Design of an Electronic Computing Instrument. In: W. Aspray & A. Burks (Eds.), *Papers of John von Neumann on Computing and Computer Theory* (Vol. 12, pp. 97–142). MIT Press. <https://archive.org/details/papersofjohnvonn00vonn>. (The Institute for Advanced Study, 2 September 1947).

- [15] Cajori, F. (1993). *A History of Mathematical Notations*. New York: Dover Publications, Inc. (Two volumes bound as one. Volume I: Notations in Elementary Mathematics. Volume II: Notations Mainly in Higher Mathematics)
- [16] Cantor, M. (1865). Über einen Codex des Klosters Salem. *Zeitschrift für Mathematik und Physik*, 10, 1–16. <https://archiv.ub.uni-heidelberg.de/volltextserver/12869/>.
- [17] Caramuelis, I. (1670). *Mathesis biceps vetus et nova*. Officinâ Episcopali. [https://books.google.pl/books?id=KRtetV1MJnkC&printsec=frontcover&source=gbs\\_book\\_other\\_versions\\_r&redir\\_esc=y#v=onepage&q&f=false](https://books.google.pl/books?id=KRtetV1MJnkC&printsec=frontcover&source=gbs_book_other_versions_r&redir_esc=y#v=onepage&q&f=false).
- [18] Chaitin, G. (2007). Epistemology as Information Theory: From Leibniz to  $\Omega$ . In: G. D. Crnkovic (Ed.), *Computation, Information, Cognition: The Nexus and the Liminal* (p. 2–17). Newcastle UK: Cambridge Scholar Publishing.
- [19] Chaitin, G. J. (2004). Leibniz, Randomness and the Halting Probability. <http://www.cs.auckland.ac.nz/CDMTCS/chaitin/turing.html>.
- [20] Chaitin, G. J. (2005). *Meta Math! The Quest for Omega*. New York: Pantheon. <https://arxiv.org/pdf/math/0404335.pdf>.
- [21] Cherry, S. R. (2017). *The Reason of Reason: How Reason Logic and Intelligibility together are evidence for God* (2<sup>nd</sup> ed.). Canterbury: Telos Publishing.
- [22] Copeland, J., Bowen, J., Sprevak, M., & Wilson, R. (2016). Is the Whole Universe a Computer? In: J. Jack Copeland, J. Bowen, M. Sprevak, & R. Wilson (Eds.), *The Turing Guide: Life, Work, Legacy* (pp. 445–462). Oxford: Oxford University Press.
- [23] Couturat, L. (1901). *La logique de Leibniz. D'après des documents inédits*. Paris: Felix Alcan. <https://archive.org/details/lalogiquedeleib00coutgoog/page/n10/mode/2up>. (Repr. Georg Olms: Hildesheim 1961, 1969)
- [24] Davis, M. (2001). *Engines of Logic: Mathematicians and the Origin of the Computer*. New York: W. W. Norton & Company.
- [25] De Morgan, A. (1872). *A budget of paradoxes*. London: Longmans, Green, and Co. <https://archive.org/details/budgetofparadoxe00demorich>.
- [26] Digges, L. (1579). *An Arithmetical Warlike Treatise, Named Stratioticos*. London: Henrie Bynneman.
- [27] Digges, L. & Digges, T. (1590). *An Arithmetical Warlike Treatise Named Stratioticos* (see (Digges, 1579) ed.). London: Richard Field.
- [28] Dodig-Crnkovic, G. (2013). Alan Turing's Legacy: Info-Computational Philosophy of Nature. In: G. Dodig-Crnkovic & R. Giovagnoli (Eds.), *Computing Nature* (pp. 115–123). Berlin-Heidelberg: Springer-Verlag.
- [29] Du Bois-Reymond, E. H. (1872). Über die Grenzen des Naturerkennens: Ein Vortrag in der zweiten öffentlichen Sitzung der 45. Versammlung deutscher Naturforscher und Ärzte zu Leipzig am 14. August 1872. Leipzig: Von Veit & Co.
- [30] Du Bois-Reymond, E. H. (1882). Über die Grenzen des Naturerkennens. *Die Sieben Welträthsel*. Leipzig: Von Veit & Co. <https://wellcomelibrary.org/item/b28103555#?c=0&m=0&s=0&cv=8&z=-1.0775%2C0.2077%2C2.7482%2C1.7807>.
- [31] Eddington, A. S. (2014). *The Nature of the Physical World: Gifford Lectures of 1927*. Cambridge: Cambridge Scholars Publishing. (Annotated and Introduced by H. G. Callaway).
- [32] Fairweather, E. R. (1956). *A Scholastic Miscellany: Anselm to Ockham*. Philadelphia: Westminster Press.
- [33] Fibonacci, Leonardus. (1202). *Liber abbaci*. [https://la.wikisource.org/wiki/Liber\\_abbaci](https://la.wikisource.org/wiki/Liber_abbaci).

- [34] Fitzpatrick, R. (Ed.). (2008). *Euclid's elements of geometry* (First edition – 2007 Revised and corrected – 2008 ed.). <https://farside.ph.utexas.edu/Books/Euclid/Elements.pdf>. (The Greek text of J.L. Heiberg (1883–1885) from *Euclidis Elementa*, edidit et Latine interpretatus est I. L. Heiberg, in aedibus B. G. Teubneri, 1883–1885; edited, and provided with a modern English translation, by Richard Fitzpatrick).
- [35] Floridi, L. (2008). A Defense of Informational Structural Realism. *Synthese*, 161(2), 219–253.
- [36] Friedlein, G. (1861). *Gerbert, die Geometrie des Boethius und die indischen Ziffern: Ein Versuch in der Geschichte der Arithmetik*. Erlangen: Verlag von Theodor Blaesing. <https://ia800205.us.archive.org/28/items/gerbertdiegeome01dostgoog/gerbertdiegeome01dostgoog.pdf>.
- [37] Fuller, R. B. (1982). *Critical path*. San Francisco, CA: Estate of R. Buckminster Fuller. (Contributor: Kiyoshi Kuromiya).
- [38] Galileo Galilei. (1623). *The Assayer* (Il Saggiatore). Stanford University. <https://web.stanford.edu/~jsabol/certainty/readings/Galileo-Assayer.pdf>. (abridged, translation by Stillman Drake).
- [39] Gerritzen, L. (2008). Warum wir Zahlen von hinten nach vorne lesen und warum das nicht so bleiben muss. In L. Gerritzen (Ed.), *Zwanzigeins: Für die unverdrehte Zahlensprechweise. Fakten, Argumente, Meinungen* (pp. 22–33). Bochum: Brockmeyer Verlag. [https://zwanzigeins.jetzt/downloads/Gerritzen%20et%20al%20\(2008\)%20-%20Zwanzigeins.pdf](https://zwanzigeins.jetzt/downloads/Gerritzen%20et%20al%20(2008)%20-%20Zwanzigeins.pdf).
- [40] Guthrie, K. S., & Fidler, D. (1987). *The Pythagorean Sourcebook: An Anthology of Ancient Writings which Relate to Pythagoras and Pythagorean Philosophy*. Gloucester, UK: Phanes Press. (Compiled & translated by Kenneth Sylvan Guthrie. Edited & introduced by David Fidler).
- [41] Hall, A. R. (1956). *The Scientific Revolution, 1500–1800: The Formation of the Modern Scientific Attitude*. Boston: Beacon Press. <https://archive.org/details/scientificrevolu00hall>.
- [42] Hall, A. R. (1966). *Rewolucja naukowa 1500–1800: Kształtowanie się nowożytnej postawy naukowej*. Warszawa: IW Pax. (Przełożył T. Zembrzuski).
- [43] Hawking, S., & Mlodinow, L. (2010). *The Grand Design*. New York: Bantam Books.
- [44] Heller, M. (2013). *Bóg i nauka: Moje dwie drogi do jednego celu*. Kraków: Copernicus Center Press. (E. Nicewicz-Staszowska (tłum.)).
- [45] Heller, M. (2014). *Granice nauki*. Kraków: Copernicus Center Press.
- [46] Hergenhahn, R. (2008). Die Köbelschen Zahlentafeln in seinen Rechenbüchern. In L. Gerritzen (Ed.), *Zwanzigeins: Für die unverdrehte Zahlensprechweise. Fakten, Argumente, Meinungen* (pp. 109–112). Bochum: Brockmeyer Verlag. [https://zwanzigeins.jetzt/downloads/Gerritzen%20et%20al%20\(2008\)%20-%20Zwanzigeins.pdf](https://zwanzigeins.jetzt/downloads/Gerritzen%20et%20al%20(2008)%20-%20Zwanzigeins.pdf).
- [47] Hilbert, D. (1900). *Mathematische Probleme*. *Nachrichten von der Königlichen Gesellschaft der Wissenschaften zu Göttingen, Math.-Phys. Klasse*(3), 253–297.
- [48] Hilbert, D. (1935). *Naturerkennen und Logik*. In David Hilbert: *Gesammelte Abhandlungen* (Vol. 3, pp. 378–387). Berlin: Verlag von Julius Springer. <https://gdz.sub.uni-goettingen.de/id/PPN237834022>.
- [49] Hochstetter, E., Greve, H. J., & Gumin, H. (1979). *Herrn von Leibniz' Rechnung mit Null und Eins*. Siemens-Aktien-Ges., [Abt. Verlag].
- [50] Hodges, A. (1997). *Turing: A Natural Philosopher*. London: Phoenix.
- [51] Ineichen, R. (2008). Leibniz, Caramuel, Harriot und das Dualsystem. *Mitteilungen der deutschen Mathematiker-Vereinigung*, 16(1), 12–15.
- [52] Infeld, L. (1980). *Quest: An Autobiography*. London: Chelsea Publishing Company.
- [53] Jones, S. H., Adams, T. E., & Ellis, C. (Eds.). (2016). *Handbook of Autoethnography*. London and New York: Routledge.

- [54] Jordan, L. (1905). *Materialien zur Geschichte der arabischen Zahlzeichen in Frankreich*. *Archiv für Kulturgeschichte*, 3(2), 155–195. <https://archive.org/details/materialenzurges00jord/mode/2up>.
- [55] Knuth, D. E. (1997). *The Art of Computer Programming (Vol. I Fundamental algorithms)*. Boston: Addison-Wesley. <https://archive.org/details/B-001-001-249>.
- [56] Kopania, J. (2018). Leibniz i jego Bóg. Rozważania z Voltaire’em w tle. *Studia z Historii Filozofii*, 3(9), 69–101.
- [57] Krajewski, S. (2012). The Ultimate Strengthening of Turing’s Test? *Semiotica*, 184(1/4), 203–218.
- [58] Krajewski, S. (2020). On the Anti-Mechanist Arguments based on Gödel’s Theorem. *Studia Semiotyczne*, 34(1), 9–56. doi: 10.26333/sts.xxxiv1.02.
- [59] Kuhn, T. S. (1962). *The Structure of Scientific Revolutions*. Chicago: The University of Chicago Press. (polski przekład: (Kuhn, 1968)).
- [60] Kuhn, T. S. (1968). *Struktura rewolucji naukowych*. Warszawa: PWN. [fem.put.poznan.pl/poli.../4891524Kuhn%20strony%20A.docx](http://fem.put.poznan.pl/poli.../4891524Kuhn%20strony%20A.docx). ((Kuhn, 1962): przekład z angielskiego Heleny Ostromęckiej. Tłumaczenie przejrzał, redagował i posłowiem zaopatrzył Stefan Amsterdamski.).
- [61] Kuhn, T. S. (2011). *Struktura rewolucji naukowych*. Warszawa: Aletheia. (Tłumacz: Helena Ostromęcka).
- [62] Kurzweil, R. (2005). *The Singularity is Near: When Humans Transcend Biology*. New York: Viking. [https://tantor-site-assets.s3.amazonaws.com/...Singularity/B0183\\_Singularity\\_PDF\\_1.pdf](https://tantor-site-assets.s3.amazonaws.com/...Singularity/B0183_Singularity_PDF_1.pdf).
- [63] Leibniz, G. W. (1666). *Dissertatio de arte combinatoria*. Lipsiae: Joh. Simon. Fickium et Joh. Polycarp. Senboldum. [abirintoermetico.com/12ArsCombinatoria/Leibniz\\_G\\_W\\_Dissertatio\\_de\\_Arte\\_combinatoria.pdf](http://abirintoermetico.com/12ArsCombinatoria/Leibniz_G_W_Dissertatio_de_Arte_combinatoria.pdf), <https://archive.org/details/ita-bnc-mag-00000844-001/page/n11/mode/2up>.
- [64] Leibniz, G. W. (1679). *De progressionē dyadica (Vol. Pars I)*. (Published in facsimile (with German translation) in (Hochstetter, Greve, & Gumin, 1979)).
- [65] Leibniz, G. W. (1697). *Brief an den Herzog von Braunschweig-Wolfenbüttel Rudolph August, 2. Januar 1697*. <http://www.fh-augsburg.de/~harsch/germanica/Chronologie/17Jh/Leibniz/lei-bina.html>.
- [66] Leibniz, G. W. (1890a). *Dialogus*. In C. I. Gerhardt (Ed.), *Die philosophischen Schriften von G. W. Leibniz (Vol. 7, pp. 190–193)*. Berlin. (Reprint: Hildesheim 1960).
- [67] Leibniz, G. W. (1890b). *Philosophische Schriften (Vol. 7; C. I. Gerhardt, Ed.)*. Berlin: Weidmann.
- [68] Leibniz, G. W. (1929). *Machina arithmetica in qua non additio tantum et subtractio sed et multiplicatio nullo, divisio vero paene nullo animi labore peragantur*. In D. E. Smith (Ed.), *A source book in mathematics (1st ed., pp. 173–181)*. New York: McGraw Hill Book Company. <https://archive.org/details/sourcebookinmath00smit/mode/2up>.
- [69] Leibniz, G. W. (1990). *Leibniz korrespondiert mit China: der Briefwechsel mit den Jesuitenmissionaren (1689–1714)* (R. Widmaier, Ed.). Frankfurt am Main: V. Klostermann.
- [70] Lesne, A. (2007). *The Discrete versus Continuous Controversy in Physics*. *Mathematical Structure in Computer Science*(17), 185–223.
- [71] Ligonnière, R. (1992). *Prehistoria i historia komputerów*. Wrocław: Ossolineum.
- [72] Marciszewski, W. (Ed.). (1981). *Dictionary of Logic as Applied in the Study of Language: Concepts/Methods/Theories*. The Hague-Boston-London: Martinus Nijhoff Publishers.

- [73] Marciszewski, W. & Stacewicz, P. (2011). *Umysł-komputer-Świat: O zagadce umysłu z informatycznego punktu widzenia* (prof. Leonard Bolc, Ed.). Warszawa: Akademicka Oficyna Wydawnicza EXIT.
- [74] Martin, M., & Augustine, K. (Eds.). (2015). *The Myth of an Afterlife: The Case against Life after Death*. Lanham, MD: Rowman & Littlefield Publishers.
- [75] Mazur, J. C. (2014). *A Short History of Mathematical Notation and its Hidden Powers*. Princeton: Princeton University Press.
- [76] Menninger, K. (1934). *Zahlwort und Ziffer: Aus der Kulturgeschichte unserer Zahlsprache, unserer Zahlschrift und des Rechenbrettes*. Breslau: Ferdinand Hirt.
- [77] Menninger, K. (1958). *Zahlwort und Ziffer: eine Kulturgeschichte der Zahl* (2<sup>nd</sup> ed., Vols. 1–2). Göttingen: Vandenhoeck & Ruprecht. (First edition: (Menninger, 1934))
- [78] Menninger, K. (1969). *Number Words and Number Symbols: A Cultural History of Numbers*. Cambridge, MA: M.I.T. Press. (Translation of (Menninger, 1958)).
- [79] Newton, I. (1675). Letter from sir Isaac Newton to Robert Hooke. <https://digitalibrary.hsp.org/index.php/Detail/objects/9792>. Historical Society of Pennsylvania.
- [80] Newton, I. (1687). *Philosophiæ Naturalis Principia Mathematica*. London: Jussu Societatis Regiæ ac Typis Josephi Streater. Prostat apud plures Bibliopolas. <http://www.gutenberg.org/ebooks/28233>. (Project Gutenberg).
- [81] O'Connor, J. J., & Robertson, E. F. (2012). Gerbert of Aurillac. In: *MacTutor history of mathematics*. School of Mathematics and Statistics University of St Andrews, Scotland. <https://mathshistory.st-andrews.ac.uk/Biographies/Gerbert/>.
- [82] Ostashevsky, E. (2000). *Quintessence from Nothingness: Zero, Platonism, and the Renaissance* (a dissertation for the degree of Doctor of Philosophy, Stanford University, 300 North Zeeb, Ann Arbor, MI). [https://www.academia.edu/43659945/Quintessence\\_from\\_nothingness\\_9986171\\_978\\_0\\_599\\_93154\\_1](https://www.academia.edu/43659945/Quintessence_from_nothingness_9986171_978_0_599_93154_1).
- [83] Pacioli, L. (1523). *Summa de arithmetica geometria proportioni: et proportionalita...* Paganino de Paganini. Retrieved from <https://books.google.pl/books?id=iqgPe49fhrsCSare> sberiensis, I. (1159). *Metalogicus* (J. B. Hall, Ed.). Logic Museum. [http://www.logicmuseum.com/wiki/Authors/John\\_of\\_Salisbury/Metalogicon](http://www.logicmuseum.com/wiki/Authors/John_of_Salisbury/Metalogicon).
- [84] Shannon, C. E. (1958). Von Neumann's Contributions to Automata Theory. *Bull. Amer. Math. Soc.* (3, Part 2), 123–129. <https://projecteuclid.org/euclid.bams/1183522376>.
- [85] Shirley, J. W. (1951). Binary Numeration before Leibniz. *American Journal of Physics*, 19(8), 452–454.
- [86] Sibley, A. (2013). Lessons from Augustine's *De Genesi ad Litteram*—Libri Duodecim. *Journal of Creation*, 27(2), 71–77. [https://creation.com/images/pdfs/tj/j27\\_2/j27\\_2\\_71-77.pdf](https://creation.com/images/pdfs/tj/j27_2/j27_2_71-77.pdf).
- [87] Stevin, S. (1585). *L'Arithmetique*. Leyde: De l'Imprimerie de Christophe Plantin. [https://books.google.pl/books?id=1dU5AAAACAAJ&printsec=frontcover&source=gbs\\_ge\\_summary\\_r&cad=0#v=onepage&q&f=false](https://books.google.pl/books?id=1dU5AAAACAAJ&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false).
- [88] Strogatz, S. (2019). *Infinite Powers: How Calculus Reveals the Secrets of the Universe*. Boston, Mass.: Houghton Mifflin Harcourt.
- [89] Swade, D. (2002). *The Difference Engine: Charles Babbage and the Quest to Build the First Computer*. Penguin Books.
- [90] Swetz, F. J. (2003). Leibniz, the Yijing, and the Religious Conversion of the Chinese. *Mathematics Magazine*, 76(4), 276–291.
- [91] Swift, J. (1892). *Gulliver's Travels into Several Remote Nations of the World* (D. Price, Ed.). London: George Bell and Sons. <http://www.gutenberg.org/ebooks/829>.
- [92] Swift, J. (2020). *Podróże Guliwera* (P. C. Aleksandra Sekuła, Ed.). Fundacja Nowoczesna Polska. <https://wolnelektury.pl/szukaj/?q=guliwera>.



- [93] Tegmark, M. (2008). The Mathematical Universe. *Foundations of Physics*, 38(2), 101–150.
- [94] Tegmark, M. (2014). *Our Mathematical Universe*. New York: Knopf.
- [95] Tryon, E. P. (1973). Is the Universe a Vacuum Fluctuation? *Nature*, 246, 396–397. doi: 10.1038/246396a0.
- [96] Trzęsicki, K. (1987). Rola pojęcia niebytu w twórczości matematycznej. *Idea. Studia nad strukturą i rozwojem pojęć filozoficznych*, 2, 75–85. (Czarnawska, M. and Kopania, J. (red.)).
- [97] Trzęsicki, K. (2006a). From the Idea of Decidability to the Number  $\Omega$ . *Studies in Grammar, Logic and Rhetoric*, 9(22), 73–142. <http://logika.uwb.edu.pl/studies>.
- [98] Trzęsicki, K. (2006b). Leibniza idea systemu binarnego. In: J. Kopania & H. Świączkowska (Eds.), *Filozofia i myśl społeczna XVII w.* (pp. 183–203). Białystok.
- [99] Trzęsicki, K. (2006c). Leibnizjańskie inspiracje informatyki. *Filozofia Nauki*, 55(3), 21–48.
- [100] Trzęsicki, K. (2016). Can AI be Intelligent? *Studies in Logic, Grammar and Rhetoric*, 61(48), 103–131.
- [101] Trzęsicki, K. (2020a). Idea of Artificial Intelligence. *Studia Humana*, 9(3/4), 37–65.
- [102] Trzęsicki, K. (2020b). Idea sztucznej inteligencji. *Filozofia i Nauka. Studia filozoficzne i interdyscyplinarne*, 8, 69–96. (Zeszyt monograficzny pod redakcją Małgorzaty Czarnockiej i Mariusza Mazurka).
- [103] Turing, A. M. (1936–37). On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42(Series 2), 230–265.
- [104] Turing, A. M. (1950). Computing Machinery and Intelligence. *Mind: A Quarterly Review of Psychology and Philosophy*, 59(236), 433–460. Retrieved from [www.csee.umbc.edu/courses/471/papers/turing.pdf](http://www.csee.umbc.edu/courses/471/papers/turing.pdf).
- [105] Turing, A. M. (1952). The Chemical Basis of Morphogenesis. *Philosophical Transactions of the Royal Society of London*, 237(641), 37–72.
- [106] Voigt, J. (2008). Die konsequente Zahlensprechweise in der Türkei. In: L. Gerritzen (Ed.), *Zwanzigeins: für die unverdrehte Zahlensprechweise; Fakten, Argumente, Meinungen* (pp. 113–114). Bochum: Brockmeyer Verlag. [https://zwanzigeins.jetzt/downloads/Gerritzen%20et%20al%20\(2008\)%20-%20Zwanzigeins.pdf](https://zwanzigeins.jetzt/downloads/Gerritzen%20et%20al%20(2008)%20-%20Zwanzigeins.pdf).
- [107] von Neumann, J. (1922–23). Zur Einführung der transfiniten Zahlen. *Acta Litt. Ac. Asc. Hung. Fran. Joseph.*, 1, 199–208.
- [108] von Neumann, J. (1958). *The Computer and the Brain* (1st ed.). New Haven: Yale University Press. (Third edition (August 28, 2012). Foreword by Ray Kurzweil).
- [109] von Neumann, J. (1963). The General and Logical Theory of Automata. In: A. H. Taub (Ed.), *Collected works* (Vol. V, pp. 288–328). London: Pergamon Press.
- [110] von Neumann, J., & Burks, A. W. (1966). *Theory of Self-reproducing Automata*. Urbana: University of Illinois Press.
- [111] Waerden, B. L. Van der. (1961). *Science Awakening*. New York: Oxford University Press. (English translation by Arnold Dresden, with additions of the author.).
- [112] Westfall, R. S. (1983). *Biography of Isaac Newton*. Cambridge: Cambridge University Press.
- [113] Wheeler, J. A. (1989). Information, Physics, Quantum: The Search for Links. In *Proceedings III International Symposium on Foundations of Quantum Mechanics* (pp. 354–386). Tokyo.
- [114] Whitehead, A. N. (1911). *An introduction to mathematics*. London: Williams and Norgate. <https://archive.org/details/introductiontoma00whitalia>.
- [115] Wolfram, S. (2002). *A New Kind of Science*. Champaign, IL: Wolfram Media. <http://www.wolframscience.com/nks/>.
- [116] Wouk, H. (2010). *The Language God Talks: On Science and Religion*. London: Little, Brown and Company.

- [117] Zuse, K. (1967). Rechnender Raum. *Elektronische Datenverarbeitung*, 8, 336–344. [https://www.informationphilosopher.com/solutions/scientists/zuse/Rechnender\\_Raum.pdf](https://www.informationphilosopher.com/solutions/scientists/zuse/Rechnender_Raum.pdf).
- [118] Zuse, K. (1969). *Rechnender Raum: Schriften zur Datenverarbeitung*. Braunschweig: Vieweg & Sohn. (Przekład na angielski (Zuse, 1970)).
- [119] Zuse, K. (1970, February). Calculating Space (Technical Translation AZT-70-164-GEMIT No. MA 02139.PDF). Cambridge, MA: MIT.
- [120] Zuse, K. (2010). *Der Computer: Mein Lebenswerk* (5. unveränd. Aufl. ed.). Berlin Heidelberg: Springer-Verlag. (Bearbeitet von Konrad Zuse, Friedrich L. Bauer, H. Zemanek).
- [121] Zuse, K. (2012). Nature as Computation. In: *A Computable Universe: Understanding & Exploring* (1st. re-edition written in LaTeX by A. German and H. Zenil (eds.)). World Scientific. Retrieved from: /Pobrane z: <https://pdfs.semanticscholar.org/7855/c53d983e-816765f5a6c637814768897d903b.pdf>. (Followed by an Afterword by Adrian German and Hector Zenil).

### ***The Turing Paradigm***

**Abstract:** Scientific knowledge is acquired according to some paradigm. Galileo wrote that the Book of Nature was written in mathematical language and could not be understood unless one first understood the language and recognized the characters with which it was written. It is argued that Turing planted the seeds of the algorithmic paradigm. According to the Turing Paradigm, the Book of Nature is written in algorithmic language, and science aims to learn how the algorithms change the physical, social, and human universe. Some sources of the Turing Paradigm are pointed out, and a few examples of the application of the Turing Paradigm are discussed.

**Keywords:** *Galileo Galilei, Alan Turing, Konrad Zuse, zero, Arabic numeral, paradigm, mathematics, algorithmics*



# Rozdział II

## Modele klasyfikacyjne oparte na wielowymiarowych wektorach cech

Leon Bobrowski

Wydział Informatyki Politechniki Białostockiej,  
Instytut Biocybernetyki i Inżynierii Biomedycznej im. Macieja Nałęczza PAN, Warszawa

**Streszczenie:** W opracowaniu analizowane są możliwości projektowania klasyfikatorów na podstawie zbiorów danych zawierających małą liczbę wysokowymiarowych wektorów cech. Badane są teoretyczne właściwości warstw złożonych z dużej liczby klasyfikatorów liniowych opartych na wyselekcjonowanych podziorach cech o małych licznosciach. Selekcja optymalnych podziorów cech odbywa się na podstawie maksymalizacji marginesów określanych zgodnie z normą  $L1$ . Warstwy złożone (*complex layers*) agregują klasyfikatory liniowe budowane w rozłącznych podprzestrzeniach cech o małym wymiarze. Zgodnie z założeniami teorii ergodycznej, uśrednianie po małej liczbie obiektów zastąpione tu zostaje uśrednianiem po wielu rozłącznych podziorach cech.

**Słowa kluczowe:** klasyfikatory liniowe, mała liczba wielowymiarowych wektorów cech, maksymalizacja marginesów opartych na normie  $L1$ , wypukłe i odcinkowo-liniowe (CPL) funkcje kryterialne

### 1. Wstęp

Rozpoznawanie obrazów (ang. *pattern recognition*) jest grupą metod służących wykrywaniu prawidłowości w zbiorach danych reprezentowanych tradycyjnie za pomocą dużej liczby wielowymiarowych wektorów cech [1]. Układy klasyfikujące (*klasyfikatory*) mają na celu poprawne przypisywanie numerów kategorii (*klas*) kolejnym obiektom pojawiającym się na wejściu klasyfikatora [2]. W przypadku klasyfikatora binarnego liczba klas jest równa dwa. W bardziej ogólnym przypadku liczba klas jest liczbą naturalną większą od dwu, ale skończoną [3].

W rozdziale tym analizowane są możliwości projektowania klasyfikatorów na podstawie zbiorów danych zawierających małą liczbę wysokowymiarowych wektorów cech [4]. Badane są właściwości klasyfikatorów złożonych z dużej liczby klasyfikatorów liniowych opartych na wyselekcjonowanych podziorach cech o małych licznosciach. Złożone warstwy (*complex layers*) agregują klasyfikatory liniowe, budowane

w rozłącznych podprzestrzeniach cech o małym wymiarze [5]. Zgodnie z założeniami teorii ergodycznej, uśrednianie po małej liczbie obiektów zastąpione tu zostaje uśrednianiem po wielu rozłącznych podzbiorach cech [6].

Projektowanie klasyfikatorów liniowych na podstawie małej liczby wysokowymiarowych wektorów cech odbywa się poprzez minimalizację wypukłych i odcinkowo-liniowych (ang. *convex and piecewise linear* – CPL) funkcji kryterialnych [7]. Minimalizacja funkcji kryterialnych typu CPL może być efektywnie realizowana za pomocą algorytmów wymiany rozwiązań bazowych [8]. Projektowanie klasyfikatorów liniowych oparte zostało na zasadzie maksymalizacji marginesów określonych za pomocą normy typu  $L_1$ . Maksymalizacja marginesów opartych na normie  $L_1$  umożliwia m.in. efektywną realizację procedur grupowania genów [9].

## 2. Liniowa separowalność wektorów cech

Bierzemy pod uwagę zbiór danych  $C$  zbudowany z  $m$  wektorów cech  $\mathbf{x}_j = [x_{j,1}, \dots, x_{j,n}]^T$  o wymiarze równym  $n$  [1]:

$$C = \{\mathbf{x}_j\}, \text{ gdzie } j = 1, \dots, m. \quad (1)$$

Wektor cech  $\mathbf{x}_j = [x_{j,1}, \dots, x_{j,n}]^T$  ( $j = 1, \dots, m$ ) reprezentuje  $j$ -ty obiekt  $O_j$  zawarty w bazie danych.

Składowe  $x_{j,i}$  wektora cech  $\mathbf{x}_j$  są liczbowymi wynikami pomiarów  $n$  cech  $X_i$  obiektu  $O_j$ . Zakładamy, że wartościami poszczególnych cech  $X_i$  ( $i = 1, \dots, n$ ) mogą być liczby rzeczywiste ( $x_{j,i} \in R^1$ ) lub binarne ( $x_{j,i} \in \{0,1\}$ ).

Poszczególne wektory cech  $\mathbf{x}_j$  ( $j = 1, \dots, m$ ) mogą być interpretowane jako punkty w  $n$ -wymiarowej przestrzeni cech  $F[n]$  ( $\mathbf{x}_j \in F[n]$ ). Bierzemy pod uwagę przypadek, gdy liczba  $n$  cech  $X_i$  jest znacznie większa niż liczba  $m$  wektorów  $\mathbf{x}_j$  ( $n \gg m$ ) [4]. Tego typu sytuację mamy np. w przypadku zbiorów danych genetycznych. W tym przypadku składowe  $x_{j,i}$  wektora cech  $\mathbf{x}_j$  mogą reprezentować aktywność poszczególnych genów  $X_i$ .

W celu budowy  $k$ -tego modelu klasyfikacyjnego zbiór danych  $C$  (1) jest dzielony zgodnie z wiedzą ekspertów na dwa podzbiory uczące  $\mathbf{G}_k^+$  i  $\mathbf{G}_k^-$ :

$$\mathbf{G}_k^+ = \{\mathbf{x}_j; j \in J_k^+\}, \text{ and } \mathbf{G}_k^- = \{\mathbf{x}_j; j \in J_k^-\}, \quad (2)$$

gdzie  $J_k^+$  oraz  $J_k^-$  są rozłącznymi zbiorami indeksów  $j$  wektorów cech  $\mathbf{x}_j$  ( $J_k^+ \cap J_k^- = \emptyset$ ).

Zbiór uczący  $\mathbf{G}_k^+$  (2) zawiera  $m_k^+$  wektorów cech  $\mathbf{x}_j$ , i reprezentuje kategorię dodatnią  $\omega_k^+$ , a zbiór  $\mathbf{G}_k^-$  zawiera  $m_k^-$  wektorów cech  $\mathbf{x}_j$  i reprezentuje kategorię ujemną  $\omega_k^-$ .

*Definicja 1:* Zbiory uczące  $\mathbf{G}_k^+$  i  $\mathbf{G}_k^-$  (2) są liniowo separowane wtedy i tylko wtedy, gdy istnieje rozwiązanie  $\mathbf{w}_k$  i  $\theta_k$  poniższego układu nierówności liniowych [3]:

$$\begin{aligned} (\exists \mathbf{w}_k, \theta_k) (\forall \mathbf{x}_j \in \mathbf{G}_k^+) \mathbf{w}_k^T \mathbf{x}_j \geq \theta_k + 1 \text{ oraz} \\ (\forall \mathbf{x}_j \in \mathbf{G}_k^-) \mathbf{w}_k^T \mathbf{x}_j \leq \theta_k - 1. \end{aligned} \quad (3)$$

Parametry  $\mathbf{w}_k$  i  $\theta_k$ , dające rozwiązanie układu nierówności liniowych (3), wyznaczają hiperpłaszczyznę  $\mathbf{H}(\mathbf{w}_k, \theta_k)$  z wektorem kierunkowym  $\mathbf{w}_k = [w_{k,1}, \dots, w_{k,n}]^T \in \mathbf{R}^n$  oraz progim  $\theta_k \in \mathbf{R}^1$ , która separuje zbiory  $\mathbf{G}_k^+$  i  $\mathbf{G}_k^-$  (2) w przestrzeni cech  $\mathbf{F}[n]$  [2]:

$$\mathbf{H}(\mathbf{w}_k, \theta_k) = \{\mathbf{x}: \mathbf{w}_k^T \mathbf{x} = \theta_k\}. \quad (4)$$

Wszystkie wektory cech  $\mathbf{w}_j$  ze zbioru  $\mathbf{G}_k^+$  (2) leżą po dodatniej stronie hiperpłaszczyzny separującej  $\mathbf{H}(\mathbf{w}_k, \theta_k)$  ( $\mathbf{w}_k^T \mathbf{x}_j > \theta_k$ ), a wszystkie wektory cech  $\mathbf{x}_j$  ze zbioru  $\mathbf{G}_k^-$  leżą po ujemnej stronie tej hiperpłaszczyzny ( $\mathbf{w}_k^T \mathbf{x}_j < \theta_k$ ).

Hiperpłaszczyzna  $\mathbf{H}(\mathbf{w}_k, \theta_k)$  (4) separuje zbiory  $\mathbf{G}_k^+$  i  $\mathbf{G}_k^-$  (1) z poniższym marginesem  $\delta_{L_2}(\mathbf{w}_k)$ , który bazuje na normie Euklidesowej  $\|\mathbf{w}_k\|_{L_2} = (\mathbf{w}_k^T \mathbf{w}_k)^{1/2}$  wektora kierunkowego  $\mathbf{w}_k$  (4) [1]:

$$\delta_{L_2}(\mathbf{w}_k) = 2 / \|\mathbf{w}_k\|_{L_2} = 2 / (\mathbf{w}_k^T \mathbf{w}_k)^{1/2}. \quad (5)$$

Metoda wektorów wspierających (ang. *support vector machines*) SVM jest często ze spektakularnymi sukcesami stosowana w wielu praktycznych problemach klasyfikacji [10]. Metoda SVM bazuje na wyznaczaniu takiej optymalnej hiperpłaszczyzny separującej  $\mathbf{H}(\mathbf{w}_k^*, \theta_k^*)$  (4), która charakteryzuje się największym marginesem  $\delta_{L_2}(\mathbf{w}_k^*)$  (5) opartym na geometrii euklidesowej [11]. Hiperpłaszczyzna  $\mathbf{H}(\mathbf{w}_k^*, \theta_k^*)$  (4), separująca zbiory wektorów cech  $\mathbf{x}_j$  największym marginesem  $\delta_{L_2}(\mathbf{w}_k^*)$  (5), wyznaczana jest przy użyciu programowania kwadratowego.

Ostatnio zaproponowana została metoda stosowania takiej optymalnej hiperpłaszczyzny separującej  $\mathbf{H}(\mathbf{w}_k^*, \theta_k^*)$  (4), która charakteryzuje się największym marginesem  $\delta_{L_1}(\mathbf{w}_k^*)$  opartym na normie  $\|\mathbf{w}_k\|_{L_1}$  wektora kierunkowego  $\mathbf{w}_k$  (4) [9]:

$$\delta_{L_1}(\mathbf{w}_k) = 2 / \|\mathbf{w}_k\|_{L_1} = 2 / (|\mathbf{w}_{k,1}| + \dots + |\mathbf{w}_{k,n}|). \quad (6)$$

Hiperpłaszczyzna  $\mathbf{H}(\mathbf{w}_k^*, \theta_k^*)$  (4), separująca zbiory wektorów cech  $\mathbf{x}_j$  z największym marginesem  $\delta_{L_1}(\mathbf{w}_k^*)$  (6), wyznaczana jest poprzez minimalizację perceptronowej funkcji kryterialnej z regularyzacją. Perceptronowa funkcja kryterialna jest

funkcją wypukłą i odcinkowo liniową (ang. *convex and piecewise linear*) CPL, a przy jej minimalizacji używa się algorytmu wymiany rozwiązań bazowych [3].

### 3. Hiperpłaszczyzny dualne oraz wierzchołki w przestrzeni parametrów

Poszerzone wektory cech  $y_j$  ( $y_j \in \mathbf{F}[n+1]$ ) budowane są z wektorów cech  $x_j = [x_{j,1}, \dots, x_{j,n}]^T$  (1) z uwzględnieniem podziału na zbiory uczące  $\mathbf{G}_k^+$  i  $\mathbf{G}_k^-$  (2) [1]:

$$(\forall x_j \in \mathbf{G}_k^+) y_j = [x_j^T, 1]^T, \text{ and} \quad (7)$$

$$(\forall x_j \in \mathbf{G}_k^-) y_j = -[x_j^T, 1]^T.$$

Poszerzony wektor parametrów  $v_k = [v_{k,1}, \dots, v_{k,n+1}]^T$  ( $v_k \in \mathbf{R}^{n+1}$ ) ma poniższą strukturę (4):

$$v_k = [w_k^T, -\theta_k]^T = [w_{k,1}, \dots, w_{k,n}, -\theta_k]^T. \quad (8)$$

Układ nierówności (3) definiujących separowalność liniową można przedstawić w poniższej postaci, używając wektorów poszerzonych  $y_j$  (7) oraz  $v_k$  (8):

$$(\exists v_k) (\forall j \in J_k) v_k^T y_j \geq 1, \quad (9)$$

gdzie  $J_k = J_k^+ \cup J_k^-$  (2).

Wektory poszerzone  $y_j$  (7) wyznaczają hiperpłaszczyzny dualne  $h_j^1$  w przestrzeni parametrów  $\mathbf{R}^{n+1}$ :

$$(\forall j \in J_k) h_j^1 = \{v: y_j^T v = 1\}, \quad (10)$$

gdzie  $v = [v_1, \dots, v_{n+1}]^T \in \mathbf{R}^{n+1}$ .

Każdy z  $n$  wektorów jednostkowych  $e_i$  definiuje poniższą hiperpłaszczyznę  $h_i^0$  w przestrzeni parametrów  $\mathbf{R}^{n+1}$ :

$$(\forall i = 1, \dots, n) h_i^0 = \{v: e_i^T v = 0\} = \{v: w_i = 0\}. \quad (11)$$

*Definicja 2:* Wierzchołek  $\mathbf{v}_k$  rzędu  $r_k$  ( $r_k < n$ ) w przestrzeni parametrów  $\mathbf{R}^{n-1}$  ( $\mathbf{v}_k \in \mathbf{R}^{n-1}$ ) jest ulokowany w przecięciu  $r_k$  hiperpłaszczyzn  $h_j^1$  (10) oraz  $n+1-r_k$  hiperpłaszczyzn  $h_i^0$  (11).

Wierzchołek  $\mathbf{v}_k$  rzędu  $r_k$  ( $r_k < n$ ) w przestrzeni parametrów  $\mathbf{R}^{n+1}$  ( $\mathbf{v}_k \in \mathbf{R}^{n+1}$ ) może być też zdefiniowany jako rozwiązanie poniższego układu równości liniowych [3]:

$$(\forall j \in J_k) \mathbf{v}_k^T \mathbf{y}_j = 1, \text{ oraz } (\forall i \in I_k) \mathbf{v}_k^T \mathbf{e}_i = 0, \quad (12)$$

gdzie  $J_k$  jest zbiorem  $r_k$  indeksów  $j$  takich hiperpłaszczyzn  $h_j^1$  (10), które przechodzą przez wierzchołek  $\mathbf{v}_k$ , oraz  $I_k$  jest zbiorem  $n+1-r_k$  indeksów  $i$  bazowych wektorów jednostkowych  $\mathbf{e}_i$  ( $i \in I_k$ ).

Układ równości liniowych (12) można przedstawić w poniższej formie macierzowej [3]:

$$\mathbf{B}_k^T \mathbf{v}_k = \mathbf{1}_k, \quad (13)$$

gdzie  $\mathbf{B}_k = [\mathbf{y}_1, \dots, \mathbf{y}_{r_k}, \mathbf{e}_{i(r_k+1)}, \dots, \mathbf{e}_{i(n+1)}]$ , oraz  $\mathbf{1}_k = [1, \dots, 1, 0, \dots, 0]^T$ .

Jeżeli macierz  $\mathbf{B}_k$  (13) jest nieosobliwa, to wierzchołek  $\mathbf{v}_k$  może być obliczany jako suma pierwszych  $r_k$  ( $r_k < n$ ) kolumn  $\mathbf{r}_i$  macierzy odwrotnej  $(\mathbf{B}_k^T)^{-1} = [\mathbf{r}_1, \dots, \mathbf{r}_{r_k}, \mathbf{r}_{r_k+1}, \dots, \mathbf{r}_{n+1}]$ :

$$\mathbf{v}_k = (\mathbf{B}_k^T)^{-1} \mathbf{1}_k = \mathbf{r}_1 + \dots + \mathbf{r}_{r_k}, \quad (14)$$

## 4. Perceptronowa funkcja kryterialna z regularyzacją

Perceptronowa funkcja kryterialna  $\Phi_k(\mathbf{v})$  używana jest m.in. w celu badania liniowej separowalności (9) zbiorów uczących  $\mathbf{G}_k^+$  i  $\mathbf{G}_k^-$  (2) [1]. Funkcja kryterialna  $\Phi_k(\mathbf{v})$  jest ważoną sumą perceptronowych funkcji kary  $\varphi_j(\mathbf{v})$  [3]:

$$(\forall j \in J_k \text{ (9)}) \quad 1 - \mathbf{y}_j^T \mathbf{v} \quad \text{if} \quad \mathbf{y}_j^T \mathbf{v} < 1 \quad (15)$$

$$\varphi_j(\mathbf{v}) =$$

$$0 \quad \text{if} \quad \mathbf{y}_j^T \mathbf{v} \geq 1.$$

Wypukłe i odcinkowo liniowe (CPL) funkcje kary  $\varphi_j(\mathbf{v})$  służą do wymuszania realizacji nierówności liniowej separowalności (9).

Perceptronową funkcję kryterialną  $\Phi_k(\mathbf{v})$  można przedstawić w poniższej postaci [3]:



$$\Phi_k(\mathbf{v}) = \sum_{j \in J_k(9)} \alpha_j \varphi_j(\mathbf{v}) \quad (16)$$

gdzie parametry (ceny)  $\alpha_j$  są równe  $\alpha_j = 1 / m_k^+$  dla  $\mathbf{x}_j \in \mathbf{G}_k^+$  (3) oraz  $\alpha_j = 1 / m_k^-$  dla  $\mathbf{x}_j \in \mathbf{G}_k^-$ .

Funkcja kryterialna  $\Phi_k(\mathbf{v})$  (16) jest wypukła i odcinkowo liniowa (ang. *convex and piecewise linear (CPL)*) [7]. Wartość minimalną funkcji  $\Phi_k(\mathbf{v})$  (16) typu CPL można wyznaczyć za pomocą algorytmu wymiany rozwiązań bazowych w jednym z wierzchołków  $\mathbf{v}_k$  (14) [3]:

$$(\exists \mathbf{v}_k^*) (\forall \mathbf{v} \in R^{n+1}) \Phi_k(\mathbf{v}) \geq \Phi_k(\mathbf{v}_k^*) \geq 0. \quad (17)$$

Można zauważyć też, że wartość minimalna  $\Phi_k(\mathbf{v}_k^*)$  (17) jest zawarta w przedziale [0, 1]:

$$0 \leq \Phi_k(\mathbf{v}_k^*) < 1.0. \quad (18)$$

*Twierdzenie 1:* Wartość minimalna  $\Phi_k(\mathbf{v}_k^*)$  (17) funkcji  $\Phi_k(\mathbf{v})$  (16) jest równa zeru ( $\Phi_k(\mathbf{v}_k^*) = 0$ ) wtedy i tylko, gdy zbiory uczące  $\mathbf{G}_k^+$  i  $\mathbf{G}_k^-$  (2) są liniowo separowalne (9) [7].

Wartość minimalna  $\Phi_k(\mathbf{v}_k^*)$  (17) zbliża się do jedności  $\Phi_k(\mathbf{v}_k^*) \approx 1.0$ , gdy zbiory uczące  $\mathbf{G}_k^+$  i  $\mathbf{G}_k^-$  (2) pokrywają się ( $\mathbf{G}_k^+ \approx \mathbf{G}_k^-$ ).

Regularyzowana funkcja kryterialna  $\Psi_{k,\lambda}(\mathbf{v})$  jest sumą perceptronowej funkcji kryterialnej  $\Phi_k(\mathbf{v})$  (16) oraz funkcji regularyzującej  $\Psi_k'(\mathbf{w})$  [12]:

$$\Psi_{k,\lambda}(\mathbf{v}) = \Phi_k(\mathbf{v}) + \lambda \Psi_k'(\mathbf{w}), \quad (19)$$

gdzie  $\mathbf{v} = [\mathbf{w}^T, -\theta] = [w_1, \dots, w_n, -\theta]^T \in \mathbf{R}^{n+1}$ , oraz  $\lambda \geq 0$  jest *poziomem kosztów*.

Funkcja regularyzująca  $\Psi_k'(\mathbf{w})$  określona jest za pomocą poniższego równania [12]:

$$\Psi_k'(\mathbf{w}) = \sum_{i \in \{1, \dots, n\}} \gamma_i |w_i| \quad (20)$$

gdzie  $\gamma_i$  ( $\gamma_i \geq 0$ ) jest *kosztem*  $i$ -tej cechy  $X_i$ .

Standardowe wartości parametrów  $\lambda$  oraz  $\gamma_i$  (19) są równe jedności:

$$\lambda = 1, \text{ and } (\forall i = 1, \dots, n) \gamma_i = 1. \quad (21)$$

Regularyzowana funkcja kryterialna  $\Psi_{k,\lambda}(\mathbf{v})$  (19), podobnie jak funkcja  $\Phi_k(\mathbf{v})$  (16), jest wypukła i odcinkowo liniowa (CPL) [3].

W oparciu o fundamentalne twierdzenie programowania liniowego wnioskujemy, że wartość minimalną funkcji regularyzowanej  $\Psi_{k,\lambda}(\mathbf{v})$  (19) można wyznaczyć w jednym z wierzchołków  $\mathbf{v}_k$  (14) [13]:

$$(\exists \mathbf{v}_k^*) (\forall \mathbf{v} \in R^{n+1}) \Psi_{k,\lambda}(\mathbf{v}) \geq \Psi_{k,\lambda}(\mathbf{v}_k^*) > 0. \quad (22)$$

Wartość minimalna  $\Psi_{k,\lambda}(\mathbf{v}_k^*)$  funkcji regularyzowanej  $\Psi_{k,\lambda}(\mathbf{v})$  (19) oraz wierzchołek optymalny  $\mathbf{v}_k^*$  (22) używane są m.in. w metodzie RLS (ang. *relaxed linear separability*) selekcji optymalnych zestawów cech  $X_i$  [14].

Minimalizacja funkcji typu CPL, m.in. funkcji  $\Phi_k(\mathbf{v})$  (16) oraz funkcji  $\Psi_{k,\lambda}(\mathbf{v})$  (19) może odbywać się za pomocą algorytmów wymiany rozwiązań bazowych. Algorytmy wymiany rozwiązań bazowych oparte są na transformacji Gaussa-Jordana [7]. Wysoka efektywność i precyzja algorytmów wymiany rozwiązań bazowych umożliwia używanie tych algorytmów również w przypadku dużych, wielowymiarowych zbiorów danych. Między innymi selekcja optymalnych zestawów genów  $X_i$  metodą RLS realizowana była za pomocą algorytmów wymiany rozwiązań bazowych [14].

## 5. Dwuetapowa minimalizacja regularyzowanej funkcji kryterialnej

Funkcja regularyzowana  $\Psi_{k,\lambda}(\mathbf{v})$  (19) jest sumą perceptronowej funkcji kryterialnej  $\Phi_k(\mathbf{v})$  (16) oraz funkcji regularyzującej  $\Psi_k'(\mathbf{v})$ . Zgodnie z Twierdzeniem 1, wartość minimalna  $\Phi_k(\mathbf{v}_k^*)$  (17) perceptronowej funkcji kryterialnej  $\Phi_k(\mathbf{v})$  (16) jest równa zeru wtedy i tylko wtedy, gdy zbiory uczące  $\mathbf{G}_k^+$  i  $\mathbf{G}_k^-$  są liniowo separowalne (9) [12].

$$\begin{aligned} (\text{Zbiory } \mathbf{G}_k^+ \text{ i } \mathbf{G}_k^- \text{ są liniowo separowalne}) &\Rightarrow \\ &\Rightarrow (\exists \mathbf{v}_k^* (22) \Phi_k(\mathbf{v}_k^*) = 0) \end{aligned} \quad (23)$$

Jeżeli wartość minimalna  $\Phi_k(\mathbf{v}_k^*)$  (17) perceptronowej funkcji kryterialnej  $\Phi_k(\mathbf{v})$  (16) jest równa zeru w wierzchołku  $\mathbf{v}_k^*$  (14), to wszystkie  $m$  funkcji kary  $\varphi_j(\mathbf{v})$  (15) też są równe zeru w tym wierzchołku:

$$(\forall_j \in J_k (9)) \varphi_j(\mathbf{v}_k^*) = 0. \quad (24)$$

Stąd:

$$(\forall_j \in J_k (9)) \mathbf{y}_j^T \mathbf{v}_k^* \geq 1. \quad (25)$$

Powyższe nierówności oznaczają, że wierzchołek optymalny  $\mathbf{v}_k^*$  (22) jest ułożony po stronie dodatniej wszystkich  $m$  hiperpłaszczyzn dualnych  $h_j^1$  (10). Hiperpłaszczyzny dualne  $h_j^1$  (10) tworzą ściany wielościanu wypukłego  $V_k$  w przestrzeni parametrów  $\mathbf{R}^{n+1}$  ( $\mathbf{v} \in \mathbf{R}^{n+1}$ ):

$$V_k = \{\mathbf{v}: (\forall j \in J_k) \mathbf{y}_j^T \mathbf{v} \geq 1\}. \quad (26)$$

Wielościan wypukły  $V_k$  (26) jest zbiorem niepustym, gdy zbiory uczące  $\mathbf{G}_k^+$  i  $\mathbf{G}_k^-$  (2) są liniowo separowalne (9) [7]. W tym przypadku wartość funkcji kryterialnej  $\Phi_k(\mathbf{v})$  (16) jest równa zero w każdym punkcie  $\mathbf{v}$  wielościanu  $V_k$  (26):

$$V_k = \{\mathbf{v}: \Phi_k(\mathbf{v}) = 0\}. \quad (27)$$

Niech symbol  $\mathbf{H}_k$  oznacza zbiór wszystkich wierzchołków  $\mathbf{v}_k$  (14) wielościanu  $V_k$  (26):

$$\mathbf{H}_k = \{\mathbf{v}_k: \Phi_k(\mathbf{v}_k) = 0\}. \quad (28)$$

Zbiór  $\mathbf{H}_k$  (28) zawiera skończoną liczbę wierzchołków  $\mathbf{v}_k$  (14). Każdy wierzchołek  $\mathbf{v}_k$  (14) rzędu  $r_k$  (12) należący do zbioru  $\mathbf{H}_k$  (28) jest punktem przecięcia  $r_k$  hiperpłaszczyzn  $h_j^1$  ( $j \in J_k$ ) (10) oraz  $n + 1 - r_k$  hiperpłaszczyzn  $h_i^0$  ( $i \in I_k$ ) (11).

W przypadku, gdy zbiory uczące  $\mathbf{G}_k^+$  i  $\mathbf{G}_k^-$  (2) są liniowo separowalne (9), wartość minimalna  $\Psi_k(\mathbf{v}_{k,\lambda}^*)$  (22) funkcji regularyzowanej  $\Psi_{k,\lambda}(\mathbf{v})$  (19) może być wyznaczana w dwu poniższych etapach:

*Etap I:* Minimalizacja perceptronowej funkcji kryterialnej  $\Phi_k(\mathbf{v})$  (16). Ponieważ zbiory uczące  $\mathbf{G}_k^+$  i  $\mathbf{G}_k^-$  (2) są liniowo separowalne (9), to wartość minimalna  $\Phi_k(\mathbf{v}_k^*)$  (17) jest równa zero ( $\Phi_k(\mathbf{v}_k^*) = 0$ ) na koniec tego etapu.

*Etap II:* Minimalizacja członu regularyzującego  $\Psi_k'(\mathbf{w})$  (20) przy zachowaniu ograniczenia z końca etapu pierwszego ( $\Phi_k(\mathbf{v}_k^*) = 0$ ).

Dwuetałowa realizacja zadania minimalizacji funkcji regularyzowanej  $\Psi_{k,\lambda}(\mathbf{v})$  (19) pozwala na zwiększenie efektywności procesu obliczeniowego.

Zadanie minimalizacyjne *Etapu II* można zapisać jako [12]:

$$\min \{ \Psi_k'(\mathbf{w}_k): \Phi_k(\mathbf{v}_k) = 0 \} = \min \left\{ \sum_{i \in \{1, \dots, n\}} | \mathbf{w}_{k,i} | : \mathbf{v}_k \in \mathbf{H}_k(28) \right\} \quad (29)$$

gdzie  $\Phi_k(\mathbf{v}_k)$  jest wartością perceptronowej funkcji kryterialnej  $\Phi_k(\mathbf{v})$  (16) w wierzchołku  $\mathbf{v}_k$  wielościanu  $V_k$  (26).

Wartość minimalną  $\Psi_k(\mathbf{v}_{k,\lambda}^*)$  (22) funkcji regularyzowanej  $\Psi_{k,\lambda}(\mathbf{v})$  (19) można wyznaczyć w jednym z wierzchołków  $\mathbf{v}_k$  (14) wielościanu  $V_k$  (26).

Rozwiązanie problemu minimalizacji z ograniczeniami (29) prowadzi do takiego wierzchołka optymalnego  $\mathbf{v}_k^* = [(\mathbf{w}_k^*)^T, -\theta_k^*]^T$ , który charakteryzuje się najmniejszą długością  $\|\mathbf{w}_k^*\|_{L_1}$  wektora wag  $\mathbf{w}_k^*$ , spośród wszystkich wierzchołków  $\mathbf{v}_k$  (14) wielościanu  $V_k$  (26) [12]:

$$(\forall \mathbf{v}_k \in V_k(26)) \|\mathbf{w}_k\|_{L_1} \geq \|\mathbf{w}_k^*\|_{L_1}, \quad (30)$$

gdzie  $\mathbf{v}_k = [\mathbf{w}_k^T, -\theta_k]^T$  (5) oraz  $\|\mathbf{w}_k\|_{L_1} = |\mathbf{w}_{k,1}| + \dots + |\mathbf{w}_{k,n}|$ .

Poszczególne wierzchołki  $\mathbf{v}_k$  (14) rzędu  $r_k$  ( $r_k < n$ ) w wielościanie  $R_k$  (26) różnią się zestawem wektorów jednostkowych  $\mathbf{e}_i$  ( $i \in I_k(12)$ ) zawartych w bazach  $B_k$  (13), powiązanych z tymi wierzchołkami:

$$B_k = [\mathbf{y}_1, \dots, \mathbf{y}_{r_k}, \mathbf{e}_{i(r_k+1)}, \dots, \mathbf{e}_{i(n+1)}]. \quad (31)$$

Algorytmy wymiany rozwiązań bazowych pozwalają na efektywne znajdowanie wierzchołka optymalnego  $\mathbf{v}_k^* = [(\mathbf{w}_k^*)^T, -\theta_k^*]^T$  ustalonego rzędu  $r_k$ . Wierzchołek taki jest rozwiązaniem zagadnienia minimalizacji z ograniczeniami (29). Podczas  $l$ -tego kroku algorytmu, wektor jednostkowy  $\mathbf{e}_{i(l)}$  ( $i(l) \in I_k(l)$ ) zawarty w bazie  $B_k(l)$  (31) zastąpiony jest przez nowy wektor  $\mathbf{e}_{i(k)}$  ( $i(k) \notin I_k(l)$ ) pod warunkiem, że taka zamiana prowadzi do zmniejszenia długości  $\|\mathbf{w}_k\|_{L_1}$  (30) wektora  $\mathbf{w}_k$ . *Kryterium stopu* algorytmu opiera się na powyższym warunku.

Zgodnie z transformacją Gaussa-Jordana, zamiana podczas  $l$ -tego kroku wektora jednostkowego  $\mathbf{e}_{i(l)}$  w bazie  $B_k(l)$  (31) na wektor  $\mathbf{y}_{k(l)}$  powoduje poniższe zmiany kolumn  $\mathbf{r}_i(l)$  macierzy odwrotnej  $B_k(l)^{-1} = [\mathbf{r}_1(l), \dots, \mathbf{r}_{r_k}(l), \mathbf{r}_{r_k+1}(l), \dots, \mathbf{r}_n(l)]$  [3]:

$$\mathbf{r}_{i(l)}(l+1) = (1 / \mathbf{r}_{i(l)}(l)^T \mathbf{y}_{k(l)}) \mathbf{r}_{i(l)}(l) \quad (32)$$

oraz

$$\begin{aligned} (\forall i \neq i(l)) \mathbf{r}_i(l+1) &= \mathbf{r}_i(l) - (\mathbf{r}_i(l)^T \mathbf{y}_{k(l)}) \mathbf{r}_{i(l)}(l+1) = \\ &= \mathbf{r}_i(l) - (\mathbf{r}_i(l)^T \mathbf{y}_{k(l)}) / \mathbf{r}_{i(l)}(l)^T \mathbf{y}_{k(l)} \mathbf{r}_{i(l)}(l). \end{aligned}$$

Zamiana wektora jednostkowego  $\mathbf{e}_{i(l)}$  na inny wektor jednostkowy  $\mathbf{e}_{i(k)}$  w bazie  $B_k(l)$  (31) powoduje poniższe zmiany kolumn  $\mathbf{r}_i(l) = [\mathbf{r}_{i,1}(l), \dots, \mathbf{r}_{i,n}(l)]^T$  macierzy odwrotnej  $B_k(l)^{-1}$ :

$$\mathbf{r}_{i(l)}(l+1) = (1 / \mathbf{r}_{i(l)}(l)^T \mathbf{e}_{i(k)}) \mathbf{r}_{i(l)}(l) = (1 / \mathbf{r}_{i(l),i(k)}(l)) \mathbf{r}_{i(l)}(l) \quad (33)$$

oraz

$$\begin{aligned}
 (\forall i \neq i(l)) \mathbf{r}_i(l+1) &= \mathbf{r}_i(l) - (\mathbf{r}_i(l)^T \mathbf{e}_{i(k)}) \mathbf{r}_{i(l)}(l+1) = \\
 &= \mathbf{r}_i(l) - (\mathbf{r}_i(l)^T \mathbf{e}_{i(k)}) / \mathbf{r}_{i(l)}(l)^T \mathbf{e}_{i(k)} \mathbf{r}_{i(l)}(l) = \\
 &= \mathbf{r}_i(l) - (\mathbf{r}_{i,i(k)}(l) / \mathbf{r}_{i(l),i(k)}(l)) \mathbf{r}_{i(l)}(l).
 \end{aligned}$$

Porównując zależności (32) i (33) możemy zauważyć, że zamiana wektora jednostkowego  $\mathbf{e}_{i(l)}$  na wektor cech  $\mathbf{y}_{k(l)}$  jest bardziej kosztowna niż zamiana dwu wektorów jednostkowych  $\mathbf{e}_{i(l)}$  i  $\mathbf{e}_{i(k)}$  w bazie  $\mathbf{B}_k(l)$  (31). Używanie dwuetapowej procedury minimalizacji regularyzowanej funkcji kryterialnej  $\Psi_{k,\lambda}(\mathbf{v})$  (19) może być celowe ze względu na koszty obliczeń.

Algorytmy wymiany rozwiązań bazowych oparte są na realizacji transformacji Gaussa-Jordana, w kolejnych krokach  $l$ . Skutki działania wybranego algorytmu zależą od tego jaki wektor  $\mathbf{e}_{i(l)}$  lub  $\mathbf{x}_{j(l)}$  opuszcza bazę  $\mathbf{B}_k(l)$  (31) w kroku  $l$ -tym oraz jaki wektor  $\mathbf{e}_{i(k)}$  lub  $\mathbf{x}_{j(k)}$  wchodzi do tej bazy na opróżnione miejsce. *Kryterium wyjścia* określa wektor  $\mathbf{e}_{i(l)}$  lub  $\mathbf{x}_{j(l)}$  opuszczający bazę  $\mathbf{B}_k(l)$  (31), natomiast *kryterium wejścia* określa wektor  $\mathbf{e}_{i(k)}$  lub  $\mathbf{x}_{j(k)}$  wchodzący do bazy. Różne strategie minimalizacji funkcji kryterialnych typu CPL wiążą się z wyborem *kryterium wyjścia* oraz *kryterium wejścia* [3].

## 6. Właściwości wierzchołków optymalnych

Wierzchołek optymalny  $\mathbf{v}_k^*$  wyznacza minimum globalne (22) regularyzowanej funkcji kryterialnej  $\Psi_{k,\lambda}(\mathbf{v})$  (19) określonej na  $m$  wektorach cech  $\mathbf{y}_j$  (7). Bierzemy pod uwagę przypadek taki, gdy liczba  $m$  wektorów cech  $\mathbf{y}_j$  (7) jest znacznie mniejsza niż liczba  $n$  cech  $X_i$  ( $m \ll n$ ) [4]. Można założyć w tym przypadku, że wszystkie poszerzone wektory cech  $\mathbf{y}_j$  (7) mogą zostać wprowadzone do macierzy bazowej  $\mathbf{B}_k$  (31) o poniższej strukturze [12]:

$$\mathbf{B}_k = [\mathbf{y}_1, \dots, \mathbf{y}_m, \mathbf{e}_{i(m+1)}, \dots, \mathbf{e}_{i(n)}, \mathbf{e}_{i(n+1)}]. \quad (34)$$

Możliwość wprowadzenia wszystkich wektorów  $\mathbf{y}_j$  (7) ( $m \ll n$ ) wiąże się z niezależnością  $m$  wektorów cech  $\mathbf{y}_j$  (7) w wysokowymiarowej przestrzeni cech  $F[n+1]$  [12].

Równanie bazowe (13) z powyższą macierzą  $\mathbf{B}_k$  (34) określa wierzchołek  $\mathbf{v}_k$  jaką sumę (14) pierwszych  $m$  kolumn  $\mathbf{r}_i$  macierzy odwrotnej  $(\mathbf{B}_k^T)^{-1} = [\mathbf{r}_1, \dots, \mathbf{r}_m, \dots, \mathbf{r}_n]$ :

$$\mathbf{v}_k = (\mathbf{B}_k^T)^{-1} \mathbf{I}_k = \mathbf{r}_1 + \dots + \mathbf{r}_m. \quad (35)$$

Macierz  $(\mathbf{B}_k^T)^{-1} = [\mathbf{r}_1, \dots, \mathbf{r}_m, \dots, \mathbf{r}_n]$  spełnia równanie odwrotności  $(\mathbf{B}_k^T)(\mathbf{B}_k^T)^{-1} = \mathbf{I}$ , gdzie  $\mathbf{I} = [\mathbf{e}_1, \dots, \mathbf{e}_n]$  jest macierzą jednostkową. Jeżeli  $m < n / 2$ , to z równania odwrotności dla macierzy  $\mathbf{B}_k$  (34) wynikają poniższe równania liniowe:

$$(\forall i \in \{1, \dots, m\}) (\forall i' \in \{m + 1, \dots, n\}) \mathbf{r}_i^T \mathbf{e}_{i'} = 0 \quad (36)$$

oraz

$$(\forall j \in \{1, \dots, m\}) \mathbf{r}_j^T \mathbf{y}_j = 1. \quad (37)$$

Konsekwencją równości (36) jest poniższa struktura wektorów kierunkowych  $\mathbf{r}_i$ , które są równe pierwszym  $m$  kolumnom macierzy odwrotnej  $(\mathbf{B}_k^T)^{-1} = [\mathbf{r}_1, \dots, \mathbf{r}_m, \dots, \mathbf{r}_n]$ :

$$(\forall i \in \{1, \dots, m\}) \mathbf{r}_i = [r_{i,1}, \dots, r_{i,m}, 0, \dots, 0]^T. \quad (38)$$

Zgodnie z powyższym wyrażeniem, każda z  $m$  pierwszych kolumn  $\mathbf{r}_i$  macierzy odwrotnej  $(\mathbf{B}_k^T)^{-1}$  względem bazy  $\mathbf{B}_k^T$  (34) ma ostatnie  $n - m$  ( $m \ll n$ ) składowych  $r_{i,l}$  równe zero ( $r_{i,l} = 0$ , gdzie  $l = m + 1, \dots, n$ ). Z bazą  $\mathbf{B}_k^T$  (34) związany został wierzchołek  $\mathbf{v}_k$  (35) rzędu  $m$  oraz zbiór  $I_k$  (12) indeksów  $i$  bazowych wektorów jednostkowych  $\mathbf{e}_i$  ( $i \in I_k$ ).

Wierzchołek optymalny  $\mathbf{v}_k^* = [[w_{k,1}^*, \dots, w_{k,n}^*]^T, -\theta_k^*]^T$  (5) rzędu  $m$  wyznaczany jest na podstawie  $m$  poszerzonych wektorów cech  $\mathbf{y}_j$  poprzez minimalizację (22) regularizowanej funkcji kryterialnej  $\Psi_{k,\lambda}(\mathbf{v})$  (19) określonej na tych wektorach. Wierzchołek optymalny  $\mathbf{v}_k^*$  rzędu  $m$  jest takim wektorem parametrów, który ma  $m$  składowych  $w_{k,i}^*$  różnych od zera ( $w_{k,i}^* \neq 0$ ) oraz  $n - m$  składowych równych zero ( $w_{k,i}^* = 0$ ).

Takie cechy  $X_i$ , którym zostały przyporządkowane zerowe wagi  $w_{k,i}^*$  w wierzchołku optymalnym  $\mathbf{v}_k^* = [[w_{k,1}^*, \dots, w_{k,n}^*]^T, -\theta_k^*]^T$  (22) pomijane są w  $k$ -tej podprzestrzeni wierzchołkowej  $F_k^*[m]$  związanej z tym wierzchołkiem. Zredukowane wektory cech  $\mathbf{x}_j[m]$  należące do podprzestrzeni wierzchołkowej  $F_k^*[m]$  ( $\mathbf{x}_j[m] \in F_k^*[m]$ ) mają po  $m$  składowych  $x_{j,i}$  reprezentujących cechy  $X_i$  z niezerowymi wagami  $w_{k,i}^*$  ( $w_{k,i}^* \neq 0$ ). Liczba  $m$  cech  $X_i$  podprzestrzeni wierzchołkowej  $F_k^*[m]$  wiąże się ze strukturą macierzy bazowej  $\mathbf{B}_k^*$  (34), do której zostały wprowadzone wszystkie  $m$  poszerzonych wektorów  $\mathbf{y}_j$  (7), utworzonych na bazie zbiorów uczących  $\mathbf{G}_k^+$  i  $\mathbf{G}_k^-$  (2) [1].

Cechy  $X_i$  z zerowymi wagami  $w_{k,i}^*$  (22) są pomijane, ponieważ nie wpływają na wartości kolejnych iloczynów skalarnych  $(\mathbf{v}_k^*)^T \mathbf{y}_j$ . Właściwość ta została wykorzystana w metodzie RLS (ang. *Relaxed Linear Separability*) selekcji optymalnych podzbiorów cech [14].

Niech symbol  $R_{k(l)}$  oznacza wierzchołkowy podzbiór  $m$  cech  $X_i$  z niezerowymi wagami  $w_{k(l),i}^*$  w wierzchołku optymalnym  $\mathbf{v}_{k(l)}^* = [[w_{k(l),1}^*, \dots, w_{k(l),n}^*]^T, -\theta_{k(l)}^*]^T$  (22) podczas  $l$ -tego etapu:

$$R_{k(l)} = \{X_i: w_{k(l),i}^* \neq 0\}. \quad (39)$$

Podzbiór wierzchołkowy  $R_{k(l)}^*$  (39) zawiera  $m$  cech  $X_i$  i powstaje ze zbioru wszystkich  $n$  cech  $F = \{X_1, \dots, X_n\}$  po usunięciu  $n - m$  cech  $X_i$  z zerowymi wagami  $w_{k(l),i}^*$  w wierzchołku optymalnym  $\mathbf{v}_{k(l)}^*$  (22). Zbiory cech  $R_{k(l)}^*$  (39) używane są w konstrukcji kompleksowych warstw klasyfikatorów liniowych (neuronów formalnych) (1) [5].

## 7. Kompleksowe warstwy neuronów formalnych

Reguła decyzyjna  $r(\mathbf{w}_k, \theta_k; \mathbf{x})$   $k$ -tego neuronu formalnego  $NF(\mathbf{w}_k, \theta_k)$  w warstwie formułowana jest przy użyciu parametrów  $\mathbf{w}_k$  i  $\theta_k$  hiperpłaszczyzny  $H(\mathbf{w}_k, \theta_k)$  (4) [1]:

$$\begin{aligned}
 & 1, \text{ jeżeli } \mathbf{w}_k^T \mathbf{x} \geq \theta_k \\
 r_k(\mathbf{x}) &= r(\mathbf{w}_k, \theta_k; \mathbf{x}) = & (40) \\
 & 0, \text{ jeżeli } \mathbf{w}_k^T \mathbf{x} < \theta_k.
 \end{aligned}$$

Neuron formalny  $NF(\mathbf{w}_k, \theta_k)$  jest *pobudzony* ( $r_k(\mathbf{x}) = 1$ ) wtedy i tylko wtedy, gdy wejściowy wektor cech  $\mathbf{x}$  ( $\mathbf{x} \in F[n]$ ) leży po dodatniej stronie ( $\mathbf{w}_k^T \mathbf{x} \geq \theta_k$ ) hiperpłaszczyzny  $H(\mathbf{w}_k, \theta_k)$  (4), zdefiniowanej przez wektor kierunkowy  $\mathbf{w}_k$  oraz próg  $\theta_k$ . Neuron formalny  $NF(\mathbf{w}_k, \theta_k)$  (40) może być używany jako klasyfikator liniowy lokujący wektory cech  $\mathbf{x}$  do jednej z dwu kategorii  $\omega_1$  i  $\omega_0$  zgodnie z poniższą regułą klasyfikacyjną:

$$\begin{aligned}
 & \text{jeżeli } r_k(\mathbf{x}) = 1, \text{ to } \mathbf{x} \in \omega_1 \text{ oraz} & (41) \\
 & \text{jeżeli } r_k(\mathbf{x}) = 0, \text{ to } \mathbf{x} \in \omega_0.
 \end{aligned}$$

Reguła decyzyjna  $r_k(\mathbf{x})$  (40) definiuje działanie neuronu formalnego  $NF(\mathbf{w}_k, \theta_k)$  w pełnej,  $n$ -wymiarowej przestrzeni cech  $F[n]$  ( $\mathbf{x} \in F[n]$ ). Warstwy kompleksowe budowane są z neuronów formalnych  $NF(\mathbf{w}_k[m], \theta_k)$  działających w wierzchołkowych podprzestrzeniach cech  $\mathbf{R}_k[m]$  ( $\mathbf{R}_k[m] \subset F[n]$ ) o wymiarze nie większym niż  $m$ , gdzie  $m \ll n$  [5]. Wierzchołkowe podprzestrzenie cech  $\mathbf{R}_{k(l)}^*[m]$  oparte są na wierzchołkach optymalnych  $\mathbf{w}_{k(l)}^*$  (22) powiązanych z macierzami bazowymi  $\mathbf{B}_{k(l)}^*$  (34) w kolejnych etapach  $l$ , gdzie  $l = 1, \dots, L$ .

Wierzchołek optymalny  $\mathbf{w}_{k(1)}^*$  w pierwszym etapie ( $l = 1$ ) uzyskujemy w wyniku minimalizacji (22) funkcji regularyzowanej  $\Psi_{k,\lambda}(\mathbf{v}[n+1])$  (19), określonej na  $m$  wektorach cech  $\mathbf{y}_j$  (7) należących do poszerzonej przestrzeni cech  $F[n+1]$  ( $\mathbf{y}_j \in F[n+1]$ ). Wierzchołek optymalny  $\mathbf{v}_{k(1)}^* = [(\mathbf{w}_{k(1)}^*)^T, -\theta_{k(1)}^*]^T$  (22) wyznacza  $m$  cech  $X_i$  powiązanych z niezerowymi wagami  $\mathbf{w}_{k(1),i}^*$ .

Wierzchołek optymalny  $\mathbf{w}_{k(2)}^*$  w drugim etapie ( $l = 2$ ) uzyskujemy poprzez minimalizację funkcji regularyzowanej  $\Psi_{k,\lambda}(\mathbf{v}[n+1-m])$  (19), określonej na  $m$  zredukowanych wektorach cech  $\mathbf{y}_j[n+1-m]$  ( $j = 1, \dots, m$ ) (7) należących do zredukowanej podprzestrzeni cech  $F_1[n+1-m]$  ( $\mathbf{y}_j[n+1-m] \in F_1[n+1-m]$ ). Wektory zredukowane

$y_j[n + 1 - m]$  (7) reprezentują  $n + 1 - m$  cech  $X_i$  należących do zredukowanego podzbioru cech  $F_1(n + 1 - m)$ :

$$F_1(n + 1 - m) = F(n + 1) \setminus R_{k(1)}, \quad (42)$$

gdzie  $F(n + 1) = \{X_1, \dots, X_n, X_{n+1}\}$  oraz  $R_{k(1)} = \{X_i: w_{k(1),i} \neq 0\}$  (39) jest zbiorem  $m$  cech  $X_i$  z niezerowymi wagami  $w_{k(1),i}$  w wierzchołku optymalnym  $\mathbf{w}_{k(1)}^*$  (22) podczas pierwszego etapu ( $l = 1$ ).

Zredukowany podzbiór cech  $F_1(n + 1 - m)$  (42) uzyskujemy w wyniku pominięcia w zbiorze  $n + 1$  cech  $F(n + 1)$  takich  $m$  cech  $X_i$  ( $X_i \in R_{k(1)}$ ) (39)), którym odpowiadają niezerowe składowe  $w_{k(1),i}^*$  wektora optymalnego  $\mathbf{w}_{k(1)}^* = [w_{k(1),1}^*, \dots, w_{k(1),n}^*]^T$  uzyskanego w poprzednim etapie.

W drugim etapie następuje redukcja podzbioru cech  $F_1(n + 1 - m)$  (42) do podzbioru  $F_2(n + 1 - 2m)$ :

$$F_2(n + 1 - 2m) = F_1(n + 1 - m) \setminus R_{k(2)}. \quad (43)$$

Zredukowany podzbiór cech  $F_2(n + 1 - 2m)$  (43) uzyskujemy w wyniku pominięcia w zbiorze  $F_1(n + 1 - m)$  (42) takich  $m$  cech  $X_i$  ( $X_i \in F_1(n + 1 - m)$  (42)), którym odpowiadają niezerowe składowe  $w_{k(2),i}$  drugiego wektora optymalnego  $\mathbf{w}_{k(2)}^* = [w_{k(2),1}^*, \dots, w_{k(2),n}^*]^T$ .

Postępujemy podobnie w kolejnych etapach  $l$  ( $l = 1, \dots, L$ ). Zredukowany podzbiór cech  $F_l(n + 1 - lm)$  podczas etapu  $l$ -tego (42) uzyskujemy poprzez pominięcie w zbiorze  $F_{l-1}(n + 1 - (l - 1)m)$  cech  $X_i$  z podzbioru  $R_{k(l)}$  ( $X_i \in R_{k(l)}$ ) (39)):

$$F_l(n - lm) = F_{l-1}(n - (l - 1)m) \setminus R_{k(l)}. \quad (44)$$

Zgodnie z wcześniejszym opisem, podzbiory cech  $F_l(n - lm)$  (44) zmniejszają się w każdym etapie  $l$  o  $m$  cech  $X_i$  związanych z niezerowymi wagami  $w_{k(l),i}^*$  w wierzchołku optymalnym  $\mathbf{w}_{k(l)}^* = [w_{k(l),1}^*, \dots, w_{k(l),n}^*]^T$ . Wierzchołek optymalny  $\mathbf{w}_{k(l)}^* = [w_{k(l),1}^*, \dots, w_{k(l),n}^*]^T$  jest wyznaczany podczas etapu  $l$ -tego za pomocą minimalizacji funkcji regularyzowanej  $\Psi_{k,\lambda}(\mathbf{v})$  (19), określonej na  $m$  zredukowanych wektorach cech  $y_j[n - lm]$  ( $y_j[n - lm] \in F_{k(l)}[n - lm]$ ) (7).

## 8. Agregacja klasyfikatorów w warstwie kompleksowej

Reguła decyzyjna  $r_k(\mathbf{x})$  (40) definiuje działanie klasyfikatora liniowego (neuronu formalnego) w  $k$ -tej wierzchołkowej podprzestrzeni cech  $\mathbf{R}_k[m]$ . Oznacza to m.in.,



że wejściowe wektory cech  $\mathbf{x}_j[m]$  tego klasyfikatora powstają w wyniku pominięcia składowych  $x_{j,i}$  opartych na takich cechach  $X_i$ , które zostały związane z zerowymi wagami  $w_{k(i)}^*$  ( $w_{k(i)}^* = 0$ ) w wierzchołku optymalnym  $\mathbf{v}_{k(i)}^*$  (22).

*Lemat 1:* Jeżeli istnieje taka baza  $\mathbf{B}_k$  (34) poszerzonej przestrzeni cech  $F[n+1]$ , która zawiera wszystkie  $m$  wektorów cech  $\mathbf{y}_j$  ( $\mathbf{y}_j \in F[n+1]$ ), to zbiory uczące  $\mathbf{G}_k^+$  i  $\mathbf{G}_k^-$  (2) są liniowo separowalne (3).

*Dowód:* Macierz bazowa  $\mathbf{B}_k$  (34) jest odwracalna i jednoznacznie wyznacza wierzchołek  $\mathbf{v}_k = [\mathbf{w}_k^T, -\theta_k]^T$  (8) rzędu  $m$ . Z równania bazowego (13) z macierzą  $\mathbf{B}_k$  (34) wynika, że spełniony jest poniższy układ  $m$  równań liniowych z wektorem wag  $\mathbf{w}_k$  i progami  $\theta_k$  (8):

$$(\forall \mathbf{x}_j \in \mathbf{G}_k^+) \mathbf{w}_k^T \mathbf{x}_j = \theta_k + 1 \text{ oraz} \quad (45)$$

$$(\forall \mathbf{x}_j \in \mathbf{G}_k^-) \mathbf{w}_k^T \mathbf{x}_j = \theta_k - 1.$$

Spełnienie układu równań liniowych (45) oznacza, że spełniony jest także układ nierówności liniowych (3) definiujący liniową separowalność zbiorów uczących  $\mathbf{G}_k^+$  i  $\mathbf{G}_k^-$  (2).

Nieosobliwa macierz bazowa  $\mathbf{B}_k$  (34) jednoznacznie wyznacza wierzchołek  $\mathbf{v}_k = [\mathbf{w}_k^T, -\theta_k]^T$  (8) rzędu  $m$ . Z wierzchołkiem tym związana jest hiperpłaszczyzna  $\mathbf{H}(\mathbf{w}_k, \theta_k)$  (4) separująca (45) zbiory uczące  $\mathbf{G}_k^+$  i  $\mathbf{G}_k^-$  (2). Wierzchołek  $\mathbf{v}_k = [\mathbf{w}_k^T, -\theta_k]^T = [[w_{k,1}, \dots, w_{k,n}]^T, -\theta_k]^T$  (8) rzędu  $m$  wyznacza wierzchołkową podprzestrzeń cech  $\mathbf{F}_k[m]$  o wymiarze  $m$  poprzez pominięcie  $n - m$  cech  $X_i$  ( $X_i \in R_k(l)$  (39)), o zerowych wagach  $w_{k,i}$  ( $w_{k,i} = 0$ ). *Zredukowany wektor* wag  $\mathbf{w}_k[m]$  o niezerowych składowych  $w_{k,i(l)}$  ( $w_{k,i(l)} \neq 0$ ) powstaje z wektora wag  $\mathbf{w}_k$  poprzez pominięcie  $n - m$  zerowych składowych  $w_{k,i}$  ( $w_{k,i} = 0$ ):

$$\mathbf{w}_k[m] = [w_{k,i(1)}, \dots, w_{k,i(m)}], \quad (46)$$

gdzie indeksy  $i(l)$  niezerowych składowych  $w_{k,i(l)}$  wektora  $\mathbf{w}_k[m]$  nie należą do podzbioru  $I_k$  ( $i(l) \notin I_k$ ) (12).

Podzbiór  $I_k$  (12) zawiera indeksy  $i$  bazowych wektorów jednostkowych  $\mathbf{e}_i$  ( $\mathbf{e}_i \in \mathbf{B}_k$  (34)). Poprzez zmiany podzbioru  $I_k$  (12) zmieniane są wierzchołki separujące  $\mathbf{w}_k[m]$  (46). Maksymalna liczba  $l_{\max}$  różnych wierzchołków separujących  $\mathbf{w}_k[m]$  (46) rzędu  $m$  równa jest liczbie podzbiorów  $I_k$  (12).

$$l_{\max} = n! / m! (n - m)! \quad (47)$$

Z każdym wierzchołkiem separującym  $\mathbf{w}_k[m]$  (46) możemy związać margines  $\delta_{L_1}(\mathbf{w}_k[m])$  (6) oparty na normie  $\|\mathbf{w}_k[m]\|_{L_1}$  typu  $L_1$ :

$$\delta_{L1}(\mathbf{w}_k[m]) = 2 / \|\mathbf{w}_k[m]\|_{L1} = 2 / (|\mathbf{w}_{k,i(1)}| + \dots + |\mathbf{w}_{k,i(m)}|). \quad (48)$$

Minimalizacja wypukłej i odcinkowo-liniowej funkcji regularyzującej  $\Psi_k'(\mathbf{w})$  (20) daje możliwość wyznaczenia optymalnego wierzchołka separującego  $\mathbf{w}_k[m]^*$  (45) rzędu  $m$ , który wyróżnia się minimalną długością  $\|\mathbf{w}_k[m]\|_{L1} = |\mathbf{w}_{k,i(1)}| + \dots + |\mathbf{w}_{k,i(m)}|$ :

$$(\forall \mathbf{w}_k[m] \text{ (45)}) \|\mathbf{w}_k[m]\|_{L1} \geq \|\mathbf{w}_k[m]^*\|_{L1}. \quad (49)$$

Optymalny wierzchołek separujący  $\mathbf{w}_k[m]^*$  (45) o minimalnej długości  $\|\mathbf{w}_k[m]^*\|_{L1}$  (49) charakteryzuje się maksymalnym marginesem  $\delta_{L1}(\mathbf{w}_k[m])$  (48):

$$(\forall \mathbf{w}_k[m] \text{ (45)}) \delta_{L1}(\mathbf{w}_k[m]^*) \geq \delta_{L1}(\mathbf{w}_k[m]). \quad (50)$$

Maksymalizacja marginesu  $\delta_{L1}(\mathbf{w}_k[m])$  (48) ma na celu zwiększanie siły generalizacyjnej reguł klasyfikacyjnych lub regresyjnych (prognostycznych) budowanych na podstawie zbiorów danych. Znanym przykładem tego typu postępowania jest metoda wektorów wspierających SVM (ang. *Support Vector Machines*), często i z sukcesami stosowana w rozwiązywaniu wielu praktycznych problemów [10].

W kolejnych etapach  $l$  ( $l = 1, \dots, L$ ) budowania warstwy kompleksowej wyznaczone są wierzchołki optymalne  $\mathbf{w}_{k(l)}[m]^*$  rzędu  $m$  oraz zredukowane podprzestrzenie cech  $F_{k(l)}[n - lm]$  o wymiarze  $n - lm$  (44). Neuron formalny  $NF(\mathbf{w}_{k(l)}[m]^*, \theta_{k(l)})$  jest budowany (projektowany) w zredukowanej podprzestrzeni cech  $F_{k(l)}[n - lm]$  (44) w oparciu o minimalizację regularyzowanej funkcji kryterialnej  $\Psi_{k,\lambda}(\mathbf{v}[n - lm]^*)$  (19), określonej na  $m$  zredukowanych wektorach cech  $\mathbf{y}_j[n - lm]$  ( $\mathbf{y}_j[n - lm] \in F_{k(l)}[n - lm]$ ) [5].

Wierzchołek optymalny  $\mathbf{w}_{k(l)}^*$  uzyskiwany jest w wyniku dwuetapowej minimalizacji (22) funkcji regularyzowanej  $\Psi_{k,\lambda}(\mathbf{v}[n + 1 - lm])$  (19), określonej na  $m$  zredukowanych wektorach cech  $\mathbf{y}_j[n + 1 - lm]$  (7) należących do zredukowanej podprzestrzeni cech  $F_{k(l)}[n + 1 - lm]$  ( $\mathbf{y}_j[n + 1 - lm] \in F_{k(l)}[n + 1 - lm]$ ). Zredukowane podprzestrzenie cech  $F_{k(l)}[n + 1 - lm]$  używane w kolejnych etapach  $l$  spełniają poniższe relacje:

$$F[n + 1] \supset F_{k(1)}[n + 1 - m] \supset \dots \supset F_{k(L-1)}[n + 1 - (L - 1)m]. \quad (51)$$

W kolejnych etapach  $l$  ( $l = 1, \dots, L$ ) wyznaczone są także wierzchołki optymalne  $\mathbf{w}_{k(l)}[m]^*$  (22) oraz wierzchołkowe podprzestrzenie cech  $R_{k(l)}[m]$  o wymiarze  $m$ . Wierzchołkowe podprzestrzenie cech  $R_{k(l)}[m]$  definiowane są na podstawie wierzchołków optymalnych  $\mathbf{w}_{k(l)}[m]^*$ . Wierzchołkowa podprzestrzeń cech  $R_{k(l)}[m]$  w  $l$ -tym etapie opiera się na takich  $m$  cechach  $X_i$  ( $X_i \in R_k(l)$  (39)), którym odpowiadają niezzerowe składowe (wagi)  $\mathbf{w}_{k(l),i}^*$  (22) wierzchołka optymalnego  $\mathbf{v}_{k(l)}^* = [[\mathbf{w}_{k(l),1}^*, \dots, \mathbf{w}_{k(l),n}^*]^T, -\theta_{k(l)}^*]^T$  (5) rzędu  $m$ . Z każdym wierzchołkiem optymalnym  $\mathbf{w}_{k(l)}[m]^*$  możemy związać

maksymalny margines  $\delta_{L_1}(\mathbf{w}_{k(l)}[m]^*)$  (50) oparty na normie  $\|\mathbf{w}_k[m]^*\|_{L_1}$  typu  $L_1$  (48). Maksymalizacja marginesu  $\delta_{L_1}(\mathbf{w}_{k(l)}[m]^*)$  (50) wiąże się z optymalizacją doboru zestawu wektorów jednostkowych  $\mathbf{e}_i$  w bazie  $\mathbf{B}_k$  (34) ( $\mathbf{e}_i \in \mathbf{B}_k$ ). Na podstawie relacji (51) można wykazać, że maksymalne marginesy  $\delta_{L_1}(\mathbf{w}_{k(l)}[m]^*)$  (50) rzędu  $m$  są uporządkowane w poniższy sposób [9]:

$$\delta_{L_1}(\mathbf{w}_k[m]^*) \geq \delta_{L_1}(\mathbf{w}_{k(l)}[m]^*) \geq \dots \geq \delta_{L_1}(\mathbf{w}_{k(L-1)}[m]^*). \quad (52)$$

Nierówność (52) odzwierciedla m.in. taką prawidłowość, że zmniejszanie wymiaru podprzestrzeni cech  $\mathbf{F}_{k(l)}[n-lm]$  pociąga za sobą zmniejszanie się wartości maksymalnych marginesów  $\delta_{L_1}(\mathbf{w}_{k(l)}[m]^*)$  (50) rzędu  $m$  [9].

Wierzchołkowa podprzestrzeń cech  $\mathbf{R}_{k(l)}^*[m]$  (39) jest definiowana na bazie  $m$  cech  $X_i$  tworzących *podzbiór wierzchołkowy*  $R_{k(l)}^*$  (39). Wierzchołkowa podprzestrzeń cech  $\mathbf{R}_{k(l)}[m]$  (39) o wymiarze  $m$  powstaje z podprzestrzeni cech  $\mathbf{F}_{k(l)}^*[n+1-(l-1)]$  (44) w wyniku pominięcia takich  $n+1-(l-1)$  cech  $X_i$ , które mają wagi równe zeru ( $w_{k(l),i} = 0$ ) w wierzchołku optymalnym  $\mathbf{v}_{k(l)}^* = [[w_{k(l),1}^*, \dots, w_{k(l),n}^*]^T, -\theta_{k(l)}^*]^T$  (5) rzędu  $m$ . Neuron formalny  $NF(\mathbf{w}_{k(l)}^*[m], \theta_{k(l)}^*)$  oparty na wierzchołku  $\mathbf{v}_{k(l)}^*$  w warstwie kompleksowej działa na bazie zredukowanych wektorów cech  $\mathbf{y}_{i,j}[m]$  należących do wierzchołkowej podprzestrzeni cech  $\mathbf{R}_{k(l)}^*[m]$  (39):

$$(\forall \mathbf{x} \in \mathbf{G}_k^+ \cup \mathbf{G}_k^- \text{ (2)}) \quad \mathbf{y}_{i,j}[m] \in \mathbf{R}_{k(l)}^*[m]. \quad (53)$$

Z zależności (43) wynika, że wierzchołkowe podprzestrzenie cech  $\mathbf{R}_{k(l)}[m]$  definiowane są w kolejnych etapach  $l$  ( $l = 1, \dots, L$ ) na rozłącznych podzbiórach cech  $R_{k(l)}$  (39). W rezultacie każdy z neuronów formalnych  $NF(\mathbf{w}_{k(l)}, \theta_{k(l)})$   $L$ -elementowej warstwy kompleksowej działa zgodnie z regułą decyzyjną  $r_{k(l)}(\mathbf{y}[m])$  (40) na wektorach wejściowych  $\mathbf{y}_{k(l)}[m]$  (53) należących do innej podprzestrzeni cech  $\mathbf{R}_{k(l)}[m]$  ( $\mathbf{y}_{k(l)}[m] \in \mathbf{R}_{k(l)}[m]$ ) [9]:

$$1, \text{ jeżeli } \mathbf{w}_{k(l)}[m]^T \mathbf{y}_{k(l)}[m] \geq \theta_{k(l)}$$

$$r_{k(l)}(\mathbf{y}[m]) = r(\mathbf{w}_{k(l)}[m], \theta_{k(l)}; \mathbf{y}_{k(l)}[m]) = \quad (54)$$

$$0, \text{ jeżeli } \mathbf{w}_{k(l)}[m]^T \mathbf{y}_{k(l)}[m] < \theta_{k(l)}.$$

Klasyfikator liniowy oparty na  $l$ -tym neuronie formalnym  $NF(\mathbf{w}_{k(l)}, \theta_{k(l)})$  (52) warstwy kompleksowej zdefiniowany jest za pomocą reguły klasyfikacyjnej (41):

$$\text{jeżeli } r_{k(l)}(\mathbf{y}[m]) = 1, \text{ to } \mathbf{x} \in \omega_1 \text{ oraz} \quad (55)$$

$$\text{jeżeli } r_{k(l)}(\mathbf{y}[m]) = 0, \text{ to } \mathbf{x} \in \omega_0.$$

W przypadku małej liczby  $m$  wektorów cech  $\mathbf{x}_j$  (1) o dużym wymiarze  $n$  ( $m \ll n$ ) możemy na ogół zapewnić, w wyniku odpowiedniego projektowania, to że każdy z klasyfikatorów liniowych (53) bezbłędnie rozpoznaje wszystkie wektory cech  $\mathbf{x}_j$  ze zbiorów uczących  $\mathbf{G}_k^+$  i  $\mathbf{G}_k^-$  (2) [9].

Klasyfikatory liniowe  $r_{k(l)}(\mathbf{y}[m])$  (53) tworzące warstwę kompleksową mogą zostać zagregowane np. za pomocą poniższej reguły większościowej [15]:

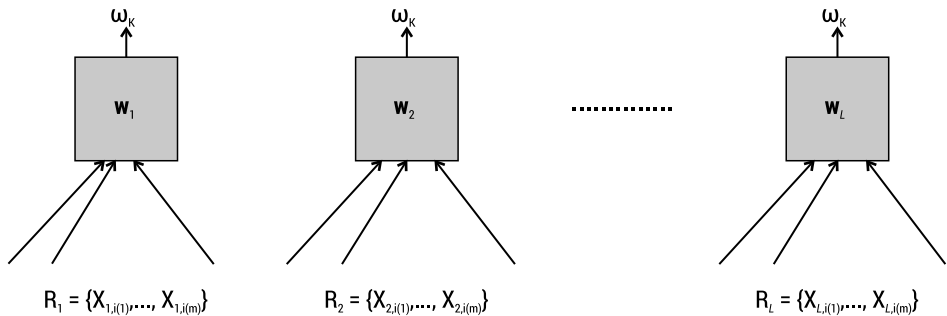
$$\text{Wektor cech } \mathbf{x} \text{ lokowany jest w klasie } \omega_1 \text{ (} \mathbf{x} \in \omega_1 \text{),} \quad (56)$$

wtedy i tylko wtedy, gdy większość z  $L$  klasyfikatorów

$r_{k(l)}(\mathbf{y}[m])$  (53) warstwy kompleksowej lokuje ten

wektor w klasie  $\omega_1$ .

Reguła decyzyjna (54) pojedynczego neuronu formalnego  $NF(\mathbf{w}_{k(l)}, \theta_{k(l)})$  jest liniowa, oparta na jednej hiperpłaszczyźnie separującej  $H(\mathbf{w}_{k(l)}, \theta_{k(l)})$  (4). Zagregowana reguła decyzyjna (56) warstwy kompleksowej zbudowanej z  $L$  ( $L > 1$ ) neuronów  $NF(\mathbf{w}_{k(l)}, \theta_{k(l)})$  nie jest liniowa, ale odcinkowo-liniowa. Reguła decyzyjna (56) oparta jest na wielościanie wypukłym ze ścianami wyznaczonymi przez  $L$  hiperpłaszczyzn separujących  $H(\mathbf{w}_{k(l)}, \theta_{k(l)})$  (4) w wierzchołkowych podprzestrzeniach cech  $\mathbf{R}_{k(l)}[m]$  (51).



RYSUNEK II.1. Przykład warstwy kompleksowej zbudowanej z  $L$  neuronów formalnych  $NF(\mathbf{w}_k, \theta_k)$  o liniowych regułach decyzyjnych (40)

Reguła decyzyjna (40) każdego z  $L$  neuronów warstwy kompleksowej (Rys. 1) zdefiniowana jest przez  $m$ -wymiarowy wektor wag  $\mathbf{w}_k[m] = [w_{k,i(1)}, \dots, w_{k,i(m)}]$  (46). Neurony formalne (elementy)  $NF(\mathbf{w}_{k(l)}, \theta_{k(l)})$  (40) warstwy kompleksowej działają w rozłącznych, wierzchołkowych podprzestrzeniach cech  $\mathbf{R}_l$  o wymiarze  $m$ . Podprzestrzenie wierzchołkowe  $\mathbf{R}_l$  definiowane są sekwencyjnie w kolejnych etapach  $l$  ( $l = 1, \dots, L$ ) na rozłącznych podzbiorach cech  $\mathbf{R}_l$  (39) zbudowanych z  $m$  cech  $X_{i(i)}$  ( $i = 1, \dots, m$ ). Liczba cech  $X_{i(i)}$  używanych w każdym klasyfikatorze  $NF(\mathbf{w}_{k(l)}, \theta_{k(l)})$  (40) warstwy jest nie większa niż liczba  $m$  wektorów cech  $\mathbf{x}_j$  w zbiorach uczących  $\mathbf{G}_k^+$  i  $\mathbf{G}_k^-$  (2) [9]. Cechy  $X_{i(i)}$  używane przez  $l$ -ty klasyfikator  $NF(\mathbf{w}_{k(l)}, \theta_{k(l)})$  (40) w warstwie nie są dostępne dla kolejnych klasyfikatorów z wyższymi wartościami  $l$ .

Każdy z klasyfikatorów liniowych  $NF(\mathbf{w}_{k(l)}, \theta_{k(l)})$  (40) warstwy kompleksowej (Rys. 1) ukierunkowany jest na rozpoznawanie tej samej kategorii  $\omega_k$ . Można zauważyć, że zgodnie z regułą klasyfikacyjną (55), wszystkie wektory cech  $\mathbf{x}_j$  ze zbiorów uczących  $\mathbf{G}_k^+$  i  $\mathbf{G}_k^-$  (2) są poprawnie klasyfikowane do kategorii pozytywnej  $\omega_k^+$  lub do kategorii negatywnej  $\omega_k^-$  przez każdy z elementów  $NF(\mathbf{w}_{k(l)}, \theta_{k(l)})$  (40) warstwy kompleksowej. Oznacza to, że jakość klasyfikacyjna warstwy kompleksowej określana jako jej *siła generalizacyjna* zależy od poprawności klasyfikacji nowych wektorów cech  $\mathbf{x}_j = [x_{j,1}, \dots, x_{j,n}]^T$  nieprzypisanych wcześniej do żadnej z kategorii  $\omega_k$ .

Zgodnie z założeniem poszczególne elementy  $NF(\mathbf{w}_{k(l)}, \theta_{k(l)})$  (40) warstwy kompleksowej projektowane są na bazie małej liczby  $m$  wektorów cech  $\mathbf{x}_j$  ze zbiorów uczących  $\mathbf{G}_k^+$  i  $\mathbf{G}_k^-$  (2). Z tego względu siła generalizacyjna pojedynczych klasyfikatorów  $NF(\mathbf{w}_{k(l)}, \theta_{k(l)})$  (40) może nie być duża. Wprowadzenie zagregowanej reguły decyzyjnej (56) daje możliwość zwiększenia siły generalizacyjnej, zwiększenia poprawności klasyfikacji nowych wektorów cech  $\mathbf{x}_j$  niezawartych w zbiorach uczących  $\mathbf{G}_k^+$  i  $\mathbf{G}_k^-$  (2).

## 9. Uwagi końcowe

Uczące zbiory danych  $\mathbf{G}_k^+$  i  $\mathbf{G}_k^-$  (2) zbudowane z małej liczby  $m$  wektorów cech  $\mathbf{x}_j$  (1) o dużym wymiarze  $n$  ( $m \ll n$ ) pojawiają się często w ważnych problemach praktycznych. Pierwszoplanowym przykładem są tu zbiory danych genetycznych [2]. W tym przypadku wektory cech  $\mathbf{x}_j$  (1) mają wymiar  $n$  większy niż 20 000, natomiast liczba  $m$  pacjentów reprezentowanych za pomocą wektorów cech  $\mathbf{x}_j$  (1) rzadko przekracza sto.

Projektowanie klasyfikatorów na podstawie uczących zbiorów danych  $\mathbf{G}_k^+$  i  $\mathbf{G}_k^-$  (2) złożonych z małej liczby  $m$  wektorów cech  $\mathbf{x}_j$  (1) o dużym wymiarze  $n$  ( $m \ll n$ ) wymaga stosowania wyspecjalizowanych metod. Uczące zbiory danych  $\mathbf{G}_k^+$  i  $\mathbf{G}_k^-$  (2) złożone z małej liczby wielowymiarowych wektorów cech  $\mathbf{x}_j$  (1) są zwykle liniowo separowalne [5]. Brak liniowej separowalności zbiorów danych  $\mathbf{G}_k^+$  i  $\mathbf{G}_k^-$  (2) może pojawić się wtedy, gdy elementy  $\mathbf{x}_j$  tych zbiorów są liniowo zależne [5]. Z tego względu duże znaczenie mają w tym przypadku klasyfikatory liniowe projektowane poprzez minimalizację wypukłych i odcinkowo liniowych (CPL) funkcji kryterialnych.

W pracy zostały przedstawione podstawy teoretyczne projektowania kompleksowych warstw klasyfikatorów liniowych (55). Kompleksowe warstwy klasyfikatorów liniowych mogą być agregowane np. zgodnie z *regułą większościową* (56). Kompleksowa agregacja klasyfikatorów liniowych pozwala na zwiększenie siły generalizacyjnej klasyfikatorów warstwowych (56) [15].

Proponowany kierunek badań dotyczy projektowania klasyfikatorów na podstawie uczących zbiorów danych  $\mathbf{G}_k^+$  i  $\mathbf{G}_k^-$  (2) obejmujących małą liczbę  $m$  wysokowymiarowych wektorów cech  $\mathbf{x}_j[n]$ , reprezentujących poszczególnych pacjentów  $P_j$ , można łączyć z elementami teorii ergodycznej [6]. Zgodnie z założeniami tej teorii proponowane jest zastąpienie uśredniania po małej liczbie  $m$  pacjentów  $P_j$ , uśrednianiem po dużej liczbie  $L$  ( $l = 1, \dots, L$ ) optymalnych podzbiorów  $F_l(n - lm)$  (44) cech (genów)  $X_l$ .

## Literatura

- [1] Duda, O. R., Hart, P. E., Stork, D. G.: *Pattern Classification*, J. Wiley, New York, 2001.
- [2] Bishop, C. M.: *Pattern Recognition and Machine Learning*, Springer Verlag, 2006.
- [3] Bobrowski, L.: *Data Exploration and Linear Separability*, pp. 1–172, Lambert, 2019.
- [4] Bobrowski, L.: "Small Samples of Multidimensional Feature Vectors" (*ICCCI 2020*), pp. 87–98 in: *Advances in Computational Collective Intelligence*, Eds.: Hernes M, et al., Springer, 2020.
- [5] Bobrowski, L.: "Complexes of Low Dimensional Linear Classifiers with  $L_1$  Margins", *ACIIDS 2021*, pp. 29–40 in: *Intelligent Information and Database Systems*, Editors: Ngoc Thanh Nguyen, et al., Springer, 2021.
- [6] Petersen, K.: *Ergodic Theory (Cambridge Studies in Advanced Mathematics)*, Cambridge University Press, 1990.
- [7] Bobrowski, L.: *Eksploracja danych oparta na wypukłych i odcinkowo-liniowych funkcjach kryterialnych*, s. 1–185, Wydawnictwa Politechniki Białostockiej, Białystok, 2005.
- [8] Bobrowski, L.: "Design of piecewise linear classifiers from formal neurons by some basis exchange technique", *Pattern Recognition*, 24(9), pp. 863–870, 1991.
- [9] Bobrowski, L., Łukaszuk T.: "Linear Classifiers with the  $L_1$  Margin from a Small Number of High-Dimensional Vectors" (12<sup>th</sup> *ACIIDS 2020*), pp. 70–89 in: *Intelligent Information and Database Systems*, Eds.: Ngoc-Thanh Nguyen, et al., Springer, 2020.
- [10] Boser, B.E., Guyon, I., & Vapnik, V.N.: "A training algorithm for optimal margin classifiers" in: *Proceedings of the Fifth Annual Workshop of Computational Learning Theory*, 5, 144–152, Pittsburgh, ACM, 1992.
- [11] Vapnik, V. N.: *Statistical Learning Theory*, John Wiley & Sons, 1998.
- [12] Bobrowski, L., Łukaszuk T.: "Repeatable Functionalities in Complex Layers of of Formal Neurons" (EANN 2021), pp. 436–447 in: *Proceedings of the 22<sup>nd</sup> Engineering Applications of Neural Networks Conference*, Eds.: Lazaros Iliadis, John Macintyre, Chrisina Jayne, Elias Pimenidis, Springer, 2021.
- [13] Simonnard, M.: *Linear Programming*, Prentice–Hall, New York, 1966.
- [14] Bobrowski, L., Łukaszuk, T.: "Relaxed Linear Separability (RLS) Approach to Feature (Gene) Subset Selection", pp. 103 – 118 in: *Selected Works in Bioinformatics*, Edited by: Xuhua Xia, INTECH, 2011.
- [15] Kuncheva, L.: *Combining Pattern Classifiers: Methods and Algorithms*, 2<sup>nd</sup> Edition, J. Wiley, New Jersey, 2014.

### ***Classification models based on multidimensional feature vectors***

**Abstract:** The study concerns the design of classifiers based on a small number of multidimensional feature vectors. The properties of layers consisting of linear classifiers aggregating a small number of features are investigated. The selection of feature subsets was based on the maximization of the  $L_1$  norm margins. The complex layer aggregates linear classifiers built on a small number of features. According to the ergodic theory assumptions, averaging a small number of objects has been replaced by averaging with many disjoint subsets of features.

**Keywords:** *linear classifiers, a small number of multidimensional feature vectors, maximization of the L1 norm margins, convex and piecewise linear (CPL) criterion functions*

Badania zostały zrealizowane w ramach pracy nr WZ/WI-IIT/3/2020 w Politechnice Białostockiej i sfinansowane z subwencji badawczej przekazanej przez ministra właściwego do spraw nauki.

# Rozdział III

## Predicting the readmission risk of heart failure patients based on claims data

Tomasz Łukaszuk\*, Jerzy Krawczuk\*, Mariusz Ferenc\*\*

\*The Faculty of Computer Science, Białystok University of Technology, Białystok, Poland

\*\*The Institute of Biocybernetics and Biomedical Engineering, PAS, Warsaw, Poland

**Abstract:** With the increased use of medical data registers, data mining has allowed for improved treatment planning and patient survival. Machine learning techniques are becoming more popular in the probability estimation for hospital patient readmission. The proper indication of high-risk patients can help reduce readmission levels. The paper aims to show how readmission risk estimation can be conducted and what results can be expected in a large population of heart failure episodes in hospital care. The problem is addressed, and its importance is investigated in an empirical study of four classifiers and several methods of diagnosis code encoding. The results show that the performance of machine learning methods measured by the area under the receiver operating characteristic curve (ROC-AUC) for heart failure readmission can be as high as 0.675. The results have been obtained on a vast population (over 600k episodes of care) using only data available on medical claims, which mainly include diagnosis codes in the Tenth Revision of the International Statistical Classification of Diseases and Related Health Problems (ICD-10). This study shows that combining patient's medical history with the patient's current episode data can improve results by over 3%. The Light GBM classifier can be trained quickly on such great data sets and provides the best results. Those results are slightly better than logistic regression (only by approximately 0.7%), but the learning time is ten times shorter.

**Keywords:** *hospital readmission, heart failure, risk prediction, machine learning, area under receiver operating characteristic curve, LightGBM, random forest, logistic regression*

## 1. Introduction

Readmission is considered when a patient discharged from a hospital is admitted again to the same or another hospital within a specific period. Readmissions are a relatively common and costly problem for hospitals [1,2]. In the USA, the average 30-day readmission rate for Medicare patients ranges from about 15% to even more than 40%,



depending on the disease unit [3]. One of the instruments supporting the reduction of readmission is determining the risk of readmission at the level of an individual patient. Such an instrument makes it possible to identify high-risk patients and try to prevent readmission.

According to the World Health Organization (WHO), 31% of all deaths in the world population are due to cardiovascular diseases (CVDs). 85% of these deaths are due to heart attacks or strokes, making them the world's leading cause of death [4].

Diseases and disorders of the circulatory system are categorized under Major Diagnostic Categories (MDC) group 5. Within this broad category there are many sub-categories called Diagnosis Related Group (DRG), for instance, cardiac arrest, arrhythmia, chest pain, or heart failure.

Heart failure (HF) is getting much attention in healthcare because it is one of the leading causes of hospitalizations and readmission [5]. Many HF patients were readmitted to the hospital within 30/90 days after discharge for various reasons. Policymakers view these readmissions as a sign of the poor quality of care and a source of excessive medical spending. According to Medicare, from 2009 to 2012, the median risk-standardized 30-day readmission rate for HF was 23.0% [6]. As a result, the Centers for Medicare and Medicaid Services (CMS) have set HF as one of their essential target diseases for designing policies to reduce readmissions.

Many computational models have been proposed for readmission risk estimations that use patient data such as Electronic Health Record (EHR) and medical claims. One of the simplest model is the LACE index [7], using four variables: Length of stay (L), Acuity of the admission (A), Comorbidity of the patient (C), and the number of Emergency Department visits within the last six months before admission (E). A higher value of this index indicates a higher risk of readmission. Another simple prediction model is the HOSPITAL score that uses seven readily available clinical predictors to identify patients with a high risk of hospital readmission. Some studies have shown it to have the significant discriminative ability [8]. Except for those simple models, there is also an important line of research in exploring machine learning techniques on multidimensional patient data, such as individual diagnoses and procedures coded in the International Statistical Classification of Diseases and Related Health Problems (ICD) standard.

The contributions of this paper can be schematized as follows:

- an experimental study on predictive modeling of the readmission risk (both 30 and 90 days) of patients with heart failure has been performed;
- the performance of various machine learning models has been compared;
- the impact of data representation (the method of encoding) on the results obtained has been studied;
- the results comparable to those found in the literature but on a significantly greater dataset (600K episodes of care) and less detailed information about patients and their hospital episodes have been achieved.

The rest of the paper is organized in the following manner: in Section 2, a short overview of related works on readmission prediction using machine learning techniques is presented; Section 3 describes machine learning techniques used in the following studies; Section 4 provides a description of the datasets used and the experimental setup; in Section 5, the results of the experiments with their interpretation are presented. Finally, the concluding remarks outline the main findings of this work.

## 2. Related works

The concept of hospital readmission appeared in the medical literature in the 1950s [9]. In the following years, among the researchers of healthcare management and treatment processes, the interest in the issues of readmission reduction and predicting the risk of readmission has been gradually increasing [10–13]. Since 2007, there has been a notably high level of interest in the above issues when the readmission rate became one of the official quality measures applied by hospitals in the USA [14].

Today, many thousands of articles and studies can be found in scientific literature, mainly medical or somehow linked to medicine, directly or indirectly concerning hospital readmission. In the following review, there will only be a few examples of works provided in which, in addition to the topic of predicting hospital readmission, machine learning also plays a significant role.

The authors of the paper [15] predict readmission using supervised learning methods on the basis of great amounts (619,274 instances) of administrative information from claims data, including patient demographics, dispensed drugs, medical or surgical procedures performed, and medical diagnosis. The objective is to gain knowledge about the predictive power of the available information. The authors test several feature selection methods, and then build the Naive Bayes and the decision tree models based on the selected features. The quality of the models is assessed in a 10-fold cross-validation procedure based on the ROC-AUC score. The results achieved by the researchers are in the range of 0.63–0.67.

In the study [16], the researchers compare the effectiveness of several machine learning algorithms to predict readmission in patients with heart failure. Using detailed clinical trials of 1653 patients, the analysts compare the effectiveness of random forests (as well as RFs combined hierarchically with support-vector machines or logistic regression), boosting, and Poisson regression against traditional logistic regression. These algorithms are used to predict 30- and 180-day all-cause readmissions and those due to heart failure. The algorithms show an advantage of about 8% to about 26% (measured based on the obtained C-statistic model) of advanced models of machine learning over logistic regression in the readmission prediction. As the best result achieved, the researchers report the C-statistics value of 0.678 in the case of prediction of 30-day readmission due to heart failure.

The goal of one of the last papers on the research topic [17] is to conduct a systematic study on the development of different types of machine learning models for predicting the readmission risk of Chronic Obstructive Pulmonary Disease (COPD) patients. The authors evaluate different approaches on a real-world database containing the medical claims of 111,992 patients from the Geisinger Health System from January 2004 to September 2015. The conducted analysis shows that predictions of the performance of ROC-AUC may range from approximately 0.60 to 0.653, depending on the set of features used. In addition, complex deep learning models, in this case, cannot improve prediction efficiency.

The above examples of works, as well as many others not described here in detail [18–23], concerning readmission prediction with the use of machine learning methods, show attempts to overcome the problem with various approaches and algorithms. It is difficult to indicate which solution has an advantage over others. The common feature of most of the described studies is assessing the prediction power of the examined models using the area under the receiver operating characteristic curve (ROC-AUC) score. The ROC-AUC score is better at fair comparisons of results because, unlike the precision and recall measures used in machine learning practice, it is independent of the decision-making threshold of the model, which may be different in particular models [24]. A second important observation is that readmission prediction is not a simple task, and the ROC-AUC score value of 0.67 should be considered as high.

### 3. Methods

This section describes the four classification models used in the experiments. It starts with the classical/statistical approach logistic regression, after which the decision tree is shortly described, which is more of a machine learning approach. Decision trees are also the basis for the next two classifiers used—random forest and light gradient boosting—both of which are ensembled classifiers.

Logistic regression (LR) is a statistical model that uses a logistic function to model the relationship between one binary dependent variable  $y$  and a group of independent variables  $\mathbf{x}$  [25]. The logistic model predicts the probability of occurrence of an event by fitting data to a logit function:

$$\text{logit}(p(y = 1|\mathbf{x}; \mathbf{w})) = \mathbf{w}^T \mathbf{x} \tag{III.1}$$

where

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right) \tag{III.2}$$

The regression coefficients  $\mathbf{w}$  must be estimated from the training data. One does it using the maximum likelihood estimation (MLE) procedure, which is equivalent to minimizing the following expression:

$$-\sum_i^N \log(p(y_i|x_i; \mathbf{w})) \quad (\text{III.3})$$

When the number of independent variables  $\mathbf{x}$  is relatively large, many of them are often not necessary, or even unnecessary, to achieve a good quality LR model. Adding a normalization factor based on the L1 norm to the minimized expression leaves only more significant features in the resulting model. The expression minimized in LR with the L1 norm (LR\_L1) takes the form:

$$-\sum_i^N \log(p(y_i|x_i; \mathbf{w})) + \lambda \|\mathbf{w}\|_1 \quad (\text{III.4})$$

where  $\lambda > 0$  is a trade-off parameter. As the value of  $\lambda$  increases, the number of independent variables used by the model decreases.

In LR with the L2 norm (LR\_L2), in order to improve numerical stabilization of the parameters  $\mathbf{w}$ , the L2 norm-based regulation factor is used:

$$-\sum_i^N \log(p(y_i|x_i; \mathbf{w})) + \lambda \|\mathbf{w}\|_2 \quad (\text{III.5})$$

The decision tree (DT), or classification tree (in the case of the present paper), is a structure similar to the flow graph. In such a structure, each internal node represents a test based on a selected attribute, each branch represents a test result, and each leaf node represents a decision made after considering the value of all attributes. The paths from the root to the leaf represent the classification rules [26]. The DT building algorithm creates subsequent tree nodes each time, assessing, usually using statistical methods, the usefulness of individual attributes to discriminate against a subset of instances of the training set whose attributes meet the tests on the path to the currently created node. Despite their apparent simplicity, DTs often prove effective even in complex classification problems.

Random forest (RF) is a classification model which consists of a large number of individual DTs that operate as an ensemble. Each tree, after proper training, returns a class prediction; the class with the most votes becomes the entire model's prediction. The fundamental concept behind random forest is the wisdom of crowds. Many relatively uncorrelated trees operating in an ensemble guarantee a better result than each tree. Therefore, the most crucial part of the RF training is the selection of appropriate subsets of data used to create individual trees in the forest [27].

LightGBM (LGBM) is a gradient-boosting tree framework considered highly efficient and scalable. What distinguishes LGBM from other tree-based algorithms is the way trees are built. In LGBM, trees grow vertically, while in most other algorithms, there is horizontal growth. In each step of building a tree, the LGBM algorithm selects a leaf with a maximum drop in the loss function and adds another tree level. Practical experiments show that the LGBM algorithm can be several times faster than the existing implementations of other gradient-boosting tree algorithms [28].

Finally, it is worth mentioning one of the essential concepts of linear algebra used in machine learning, i.e., the singular value decomposition (SVD) [29]. This work uses SVD as an effective method of dimensional reduction of data.

## 4. Empirical studies

### 4.1. Dataset

The data come from the Centers for Medicare & Medicaid Services (CMS) and are Medicare claims from Limited Data Set (LDS). We chose only those patient hospital stays (episodes of care) that were assigned to one of the three diagnosis-related groups (DRG), 291, 292, or 293, were chosen. All three are for heart failure and shock; 291 have significant complications or comorbidity (MCC), 292 have complications or comorbidity (CC), and 293 are without MCC and CC. The number of episodes and the readmission level in those three DRGs are different, as shown in Figure III.1. This figure shows a different outcome for patients within 30 and 31 to 90 days after being discharged from the hospital. Deaths and readmissions were also split to show the proportion, but in the classification task, it is treated as a binary task. Either the patients experience an undesirable event such as readmission or death, or they live without readmission for the considered period. In the experiments, we used 90-day readmissions were used in a sense that readmission occurs from day 1 to 90 after a patient is discharged from the hospital. Here are presented data from days 31–90 to illustrate the patient population better.

The data covers two years, from April 1st, 2016, to March 30<sup>th</sup>, 2018. We excluded a few episodes for patients aged below 21. In total, there were 621,108 episodes of care. The resulting population is rather old (see Figure III.2). Each episode contained information about a patient, such as gender, race, age, and information about this episode, including anchor duration, primary diagnosis and procedure, and up to twenty-five other diagnoses in the ICD-10 standard. We did not have any missing data in the analyzed dataset. The characteristics of the studied episodes are included in Table III.1.

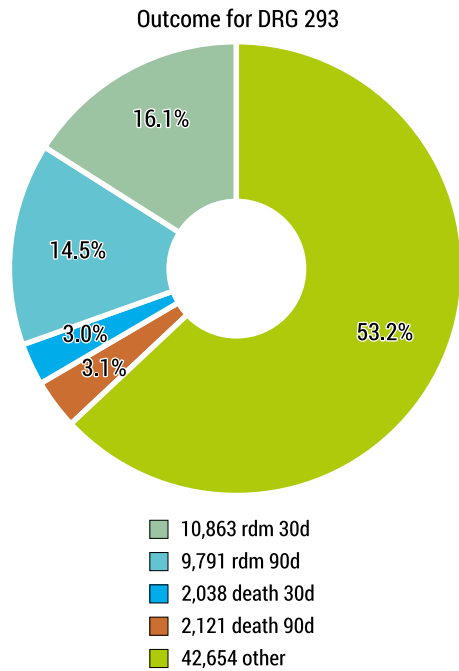
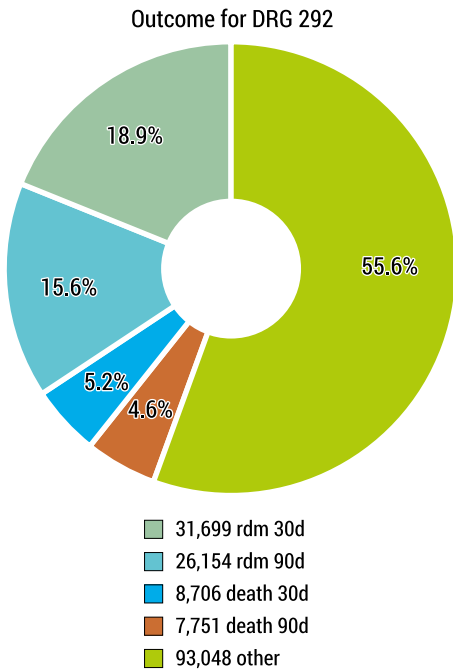
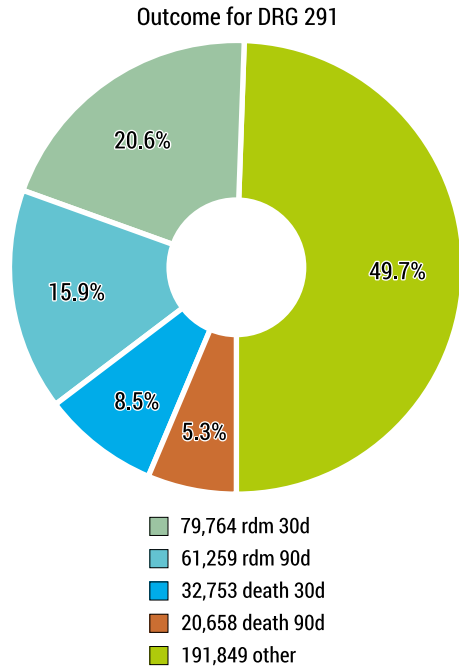
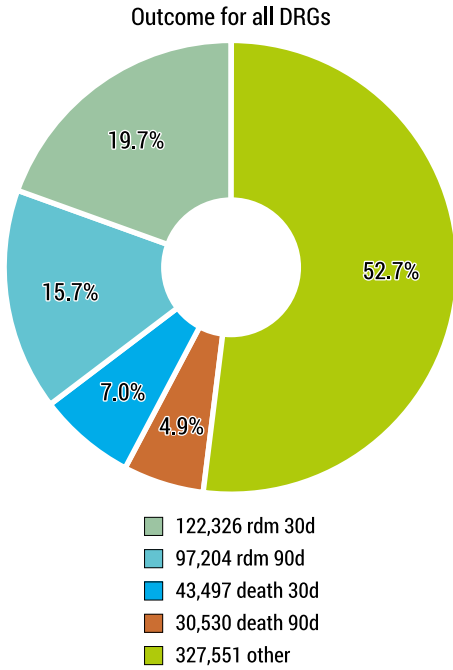


FIGURE III.1. The outcome of the episodes of care in 30 and 31–90 days after hospital discharge

TABLE III.1. Patient characteristics

Characteristic	Non-readmission	30 days readmission	31-90 days readmission
#Episodes	327,551	165,823	127,734
Age quartiles	71,79,87	70,79,87	70,80,87
Women	52.67%	50.50%	52.05%
White race	78.15%	78.57%	77.09%
Black race	15.81%	15.73%	17.20%
Anchor duration quartiles	3,4,6	3,5,7	3,5,7
IP stays 3m* no/once/more	326,224/1,239/88	164,442/1,263/118	126,878/802/54
Primary diagnosis			
I130	95,932	57,409	43,069
I110	98,506	37,434	31,688
I5033	37,887	17,772	14,392
I5023	29,194	15,983	12,251
I132	16,654	12,246	8,651
I5043	13,823	7,884	5,974
I509	12,554	7,188	4,588
I5031	10,013	3,626	2,907
I5021	6,297	2,990	1,925
I5041	2,134	874	632
Other 19 codes	4,557	2,417	1,657

\*IP stays 3m means a number of in-patients stays 3 months before hospitalization

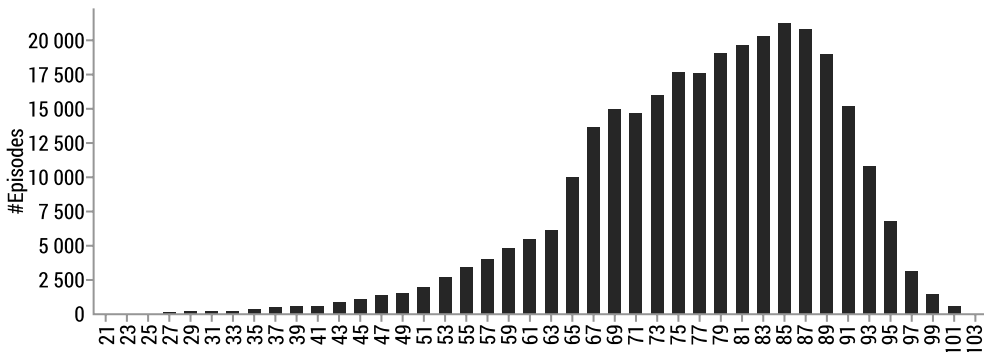


FIGURE III.2. Patient age distribution

The data encodes patient diagnoses with the International Classification of Diseases (ICD-10) codes. There is a large number of different ICD-10 codes for both current

episode diagnoses (secondary diagnosis) (13,707) and historical diagnoses (34,889). For that reason, we investigated a grouping strategy that uses a logical hierarchy of the ICD-10 codes, which allows one to group them by the first three, four, and six characters of the code. In ICD-10, the first three characters represent a category, and the next three specify the anatomic site/laterality/severity/etiology. In rare cases, there is also a seventh character specifying the visit encounter sequela typically used for injury and external cause-related codes. For instance, in ICD-10 codes, heart failure is listed in the I50 category (except for rheumatic or newborn cases):

- I50: Heart failure
- I50.1: Left ventricular failure
- I50.2: Systolic heart failure
- I50.20: Unspecified systolic heart failure
- I50.21: Acute systolic heart failure
- I50.22: Chronic systolic heart failure
- I50.23: Acute on chronic systolic (congestive) heart failure

The exclusion was performed for all primary, secondary and historical diagnoses that did not appear in at least twenty episodes; the additional one was performed after grouping 6, 4, and 3 character prefixes. Table III.2 shows how many distinct codes appear and how many of them have been excluded.

TABLE III.2. ICD-10 diagnosis codes statistics

Diagnoses	Primary	Secondary	Historical
#Distinct values for (#At least 20 episodes)			
full code	29 (25)	13,707 (4,015)	34,889 (12,953)
prefix 6	29 (25)	12,357 (3,994)	24,884 (11,707)
prefix 4	12 (12)	5,747 (2,700)	8,462 (5,526)
prefix 3	5 (5)	1,437 (996)	1,824 (1,462)
All values	621,108	10,565,776	23,633,256
Average per episode	1	17.01	38.05
#Excluded* for (Percentage of all values)			
prefix 6	43 (0.007%)	37,536 (0.36%)	74,445 (0.32%)
prefix 4	-	16,803 (0.16%)	18,385 (0.08%)
prefix 3	-	2,970 (0.03%)	2,261 (0.01%)

\*Code exclusion due to less than twenty occurrences

## 4.2. Experimental setup

To evaluate the performance, the 4-fold cross-validation is applied. In each step, 3/4<sup>th</sup> of the dataset is used for training and 1/4<sup>th</sup> for testing. In Figure III.3, the data split



is shown as the first step. This split must be done before any data transformation. The transformation is done via an encoder. For each feature, there could be a different transformation. The Min-Max scaler was used for continuous features such as age and anchor duration, but for categorical ones such as race and gender, one-hot encoding was employed. For diagnosis features represented by ICD-10 codes, we have implemented a special encoder that could be parameterized by prefix length and several dimensions for SVD transformation (with dimensions reduced to 50 and 200). As the second step after splitting the original dataset, encoders were fit only with the train set; for instance, the Min and Max values were found only in the train data. The next step was to transform both train and test sets using fitted encoders. After this step, the data is vectorized in high-dimensional space. In the fourth step, one learns the classifier on a transformed train set, and finally, the readmission is predicted on a transformed test set in the fifth step. Predictions are probabilities that one confronts with actual values. The ROC-AUC was chosen as the performance measure, which is often used if the probability is predicted, as it is in the case of this work.

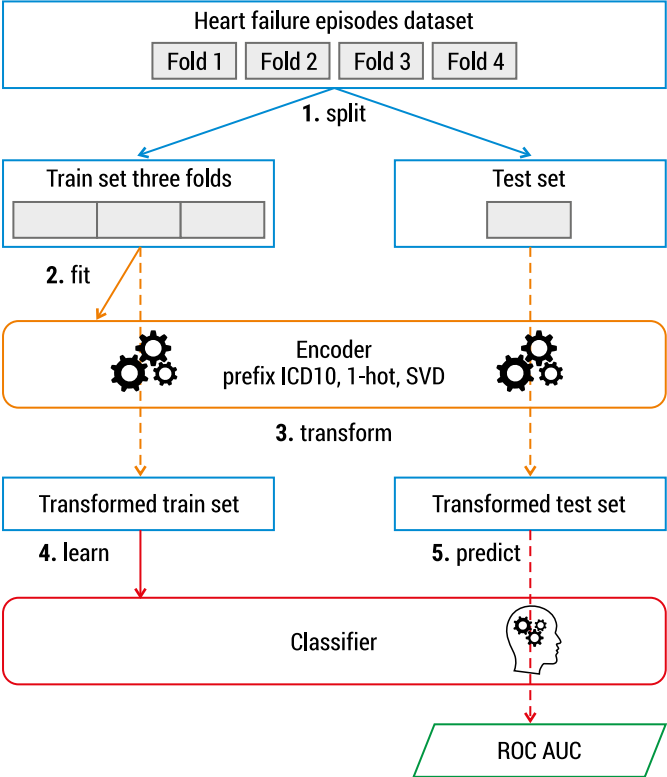


FIGURE III.3. Cross-validation loop with data transformation by an encoder. The encoder is fitted only on the training part of the data. The classifier is learned and evaluated on transformed data

## 5. Results and discussion

First, to choose what ICD-10 code prefix should be used for diagnosis in the next experiments, a series of calculations just with the Light GBM classifier was performed. This classifier can handle large data and requires lower memory to run, making it appropriate for the extensive dataset. Then, it was decided to encode the diagnoses with a one-hot encoder, using grouping codes by prefixes of 3,4, and 6 in length. In addition, raw one-hot encoding and SVD transformation to reduce dimensionality were also tested. The results are presented in Table III.3. The times in the last columns are averaged for one step of cross-validation; the first time is for one-hot encoding and SVD transformation, while the second time is for learning classifiers with all times presented in seconds. They are also averaged over calculations for 30- and 90-day readmissions. For both cases, the sets of attributes are the same with the only difference being the outcome values. All the calculations were performed on an 8-core 3GHz Intel Family 15 Model 6 processor with 32GB of RAM.

TABLE III.3. Results using only the ICD-10 diagnosis codes present in the current episode, without patient history. The prefixes 3, 4, and 6 were used and then encoded with one hot. After encoding two kinds dimensionality reductions were performed using SVD to 200 and 50 dimensions. The learned classifier LGBM with parameter estimators 1000, max depth 3. ROC-AUC is shown with a 95% confidence interval

ICD-10 prefix	Encoder	Dimension	ROC-AUC 90 days	ROC-AUC 30 days	Time
6	1hot svd no	3634	0.6543 ±0.0018	0.6603 ±0.0034	19+31
6	1hot svd 200	288	0.6474 ±0.0022	0.6529 ±0.0040	134+55
6	1hot svd 50	138	0.6370 ±0.0015	0.6420 ±0.0046	48+19
4	1hot svd no	2542	0.6544 ±0.0016	0.6518 ±0.0031	16+31
4	1hot svd 200	288	0.6477 ±0.0025	0.6467 ±0.0030	134+54
4	1hot svd 50	138	0.6372 ±0.0025	0.6366 ±0.0038	47+24
3	1hot svd no	1044	0.6500 ±0.0015	0.6254 ±0.0027	12+26
3	1hot svd 200	288	0.6471 ±0.0022	0.6240 ±0.0026	112+54
3	1hot svd 50	138	0.6371 ±0.0021	0.6244 ±0.0031	38+19

From those results, the prefix 4 for 90 days of readmission and the prefix 6 for 30 days readmission with one-hot encoding without SVD were chosen because they yielded the best results. In the next experiment, the historical diagnoses from the previous year of patient admissions have been added (see Table III.4).

TABLE III.4. The results include patient diagnosis history for up to one year before admission

ICD-10 prefix	Encoder	Dimension	ROC-AUC 90 days	ROC-AUC 30 days	Time
6	1hot svd no	14295	0.6756 ±0.0010	0.6768 ±0.0038	44+77
6	1hot svd 200	3834	0.6743 ±0.0007	0.6761 ±0.0037	175+75
6	1hot svd 50	3684	0.6735 ±0.0006	0.6760 ±0.0033	81+44
4	1hot svd no	8858	0.6728 ±0.0017	0.6757 ±0.0036	40+57
4	1hot svd 200	3834	0.6714 ±0.0016	0.6746 ±0.0037	182+105
4	1hot svd 50	3684	0.6710 ±0.0011	0.6745 ±0.0032	79+54
3	1hot svd no	5045	0.6558 ±0.0010	0.6667 ±0.0026	31+29
3	1hot svd 200	3834	0.6547 ±0.0009	0.6659 ±0.0024	149+77
3	1hot svd 50	3684	0.6546 ±0.0011	0.6656 ±0.0027	64+43

From this experiment, it can be seen that adding historical diagnoses adds 0.02 to the ROC-AUC statistic. All results with the prefix 6 are the best; using SVD makes the results slightly worse but reduces the dimensionality tremendously. The diversity of diagnoses is much higher in the patient history (10661) than in the patient heart failure episodes (3546). It was also necessary to run the last experiments with the prefix 6 and SVD 200 for the historical diagnoses in 90 days of readmission and SVD 50 for 30 days of readmission. The last experiment tests different classifiers (see Table III.5).

TABLE III.5. Results for all data, including current (prefix 4 no svd) and historical (prefix 6 and SVD 200) diagnoses

Classifier	ROC-AUC 90 days	ROC-AUC 30 days	Time
Logistic Regression L1	0.6702 ±0.0014	0.6669 ±0.0035	175+3758
Logistic Regression L2	0.6702 ±0.0014	0.6668 ±0.0035	175+1002
Decision Tree	0.6249 ±0.0022	0.6329 ±0.0009	175+402
Random Forest	0.6421 ±0.0015	0.6568 ±0.0032	175+2101
LightGBM	0.6743 ±0.0007	0.6761 ±0.0037	175+75

As seen in table 3, other classifiers do not give better results than LightGBM. Logistic regression is in second place with no significant difference, but the learning time is much longer. For 90 days of readmission, the difference is lower with LGBM outperforming the logistic regression by 0.0041, but for 30 days, the difference is higher – 0.0091. The other two classifiers give significantly worse results, especially for 90 days of readmission. The Random Forest outperforms the single decision tree by about 0.02, but it is worse than LGBM by 0.03 on 90 days of readmission and 0.02 on 30 days.

At first sight, the results of the readmission prediction do not seem spectacular. On the other hand, they are on the level of the best results obtained by the researchers

dealing with readmission prognosis in heart failure patients that were presented in the literature (see, for instance, [16, 18, 22]). However, when making comparisons, it is worth paying attention to the detail of data used to build models and required for the further prediction of readmission risk. Unfortunately, only claims data were at the disposal, which do not contain information about detailed results of tests and examinations of patients, but mainly codes of diagnosed medical disorders and medical procedures performed during hospitalization. Considering the above, the received model evaluation values should be considered very good. It is, of course, possible to build a clearly better-quality model, as in the work [23] (ROC-AUC of about 0.77), but this would require more detailed and verified information about the treatment process, including its course over time.

## 6. Concluding remarks

A series of experiments for predicting 30- and 90-day readmission risk of heart failure in patients based on their claims records has been conducted. The Authors have investigated the significance of using full diagnosis codes versus aggregating them according to the ICD-10 hierarchy and adding a singular value decomposition to reduce dimensionality. Both current and historical diagnoses and procedures were investigated. The results show that using full codes gives the best results; in some cases, it may also be worth trying prefixes of length 4. Adding historical diagnoses improves the results significantly by about 3% (0.02 ROC-AUC), and it can be achieved using the SVD transformation.

This work verifies the importance of knowing the details of patient data, including medical history, which significantly improves the results of predictive modeling of readmission.

## References

- [1] M. Zaya, A. Phan, E. R. Schwarz, The Dilemma, Causes and Approaches to Avoid Recurrent Hospital Readmissions for Patients with Chronic Heart Failure, *Heart Failure Reviews* 17 (3) (2012) 345–353.
- [2] J. S. Ross, J. Chen, Z. Lin, H. Bueno, J. P. Curtis, P. S. Keenan, S.-L. T. Normand, G. Schreiner, J. A. Spertus, M. T. Vid'an, et al., Recent National Trends in Readmission Rates after Heart Failure Hospitalization, *Circulation: Heart Failure* 3 (1) (2010) 97–103.
- [3] HospitalCompare. URL <http://www.medicare.gov/HospitalCompare/search.aspx> [Accessed: 01.06.2021].
- [4] Cardiovascular Diseases (CVDs). URL [https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)) [Accessed: 21.06.2021].
- [5] B. Ziaieian, G. C. Fonarow, The Prevention of Hospital Readmissions in Heart Failure, *Progress in Cardiovascular Diseases* 58 (4) (2016) 379–385.

- [6] D. Mozaffarian, E. J. Benjamin, A. S. Go, D. K. Arnett, M. J. Blaha, M. Cushman, S. De Ferranti, J.-P. Després, H. J. Fullerton, V. J. Howard, et al., Executive Summary: Heart Disease and Stroke Statistics–2015 Update: A Report from the American Heart Association, *Circulation* 131 (4) (2015) 434–441.
- [7] C. van Walraven, I. A. Dhalla, C. Bell, E. Etchells, I. G. Stiell, K. Zarnke, P. C. Austin, A. J. Forster, Derivation and Validation of an Index to Predict Early Death or Unplanned Readmission after Discharge from Hospital to the Community, *CMAJ* 182 (6) (2010) 551–557.
- [8] J. D. Donz’è, M. V. Williams, E. J. Robinson, E. Zimlichman, D. Aujesky, E. E. Vasilevskis, S. Kripalani, J. P. Metlay, T. Wallington, G. S. Fletcher, et al., International Validity of the HOSPITAL Score to Predict 30-Day Potentially Avoidable Hospital Readmissions, *JAMA Internal Medicine* 176 (4) (2016) 496–502.
- [9] M. Woodside, A Follow-Up of Psychiatric Patients; One Year’s Survey of Patients Discharged from the York Clinic, *Guy’s Hospital Reports* 102 (1) (1953) 70.
- [10] D. M. Smith, J. A. Norton, C. J. McDonald, Nonelective Readmissions of Medical Patients, *Journal of Chronic Diseases* 38 (3) (1985) 213–224.
- [11] J. M. Naessens, C. L. Leibson, I. Krishan, D. J. Ballard, Contribution of a Measure of Disease Complexity (COMPLEX) to Prediction of Outcome and Charges Among Hospitalized Patients, in: *Mayo Clinic Proceedings*, Vol. 67, Elsevier, 1992, pp. 1140–1149.
- [12] C. Boulton, B. Dowd, D. McCaffrey, L. Boulton, R. Hernandez, H. Krulewicz, Screening Elders for Risk of Hospital Admission, *Journal of the American Geriatrics Society* 41 (8) (1993) 811–817.
- [13] E. F. Philbin, T. G. DiSalvo, Prediction of Hospital Readmission for Heart Failure: Development of a Simple Risk Score Based on Administrative Data, *Journal of the American College of Cardiology* 33 (6) (1999) 1560–1566.
- [14] M. P. A. Commission, et al., June 2007 Report to the Congress: Promoting Greater Efficiency in Medicine (Chapter 5: Payment policy for inpatient readmissions). 2007, Washington, DC, USA.
- [15] A. Hosseinzadeh, M. Izadi, A. Verma, D. Precup, D. Buckeridge, Assessing the Predictability of Hospital Readmission Using Machine Learning, in: *Twenty-Fifth IAAI Conference*, 2013.
- [16] B. J. Mortazavi, N. S. Downing, E. M. Buchholz, K. Dharmarajan, A. Manhapra, S.-X. Li, S. N. Negahban, H. M. Krumholz, Analysis of Machine Learning Techniques for Heart Failure Readmissions, *Circulation: Cardiovascular Quality and Outcomes* 9 (6) (2016) 629–640.
- [17] X. Min, B. Yu, F. Wang, Predictive Modeling of the Hospital Readmission Risk from Patients’ Claims Data Using Machine Learning: A Case Study on COPD, *Scientific Reports* 9 (1) (2019) 2362.
- [18] J. Kerexeta, A. Artetxe, V. Escolar, A. Lozano, N. Larburu, Predicting 30-Day Readmission in Heart Failure Using Machine Learning Techniques, in: *HEALTHINF*, 2018, pp. 308–315.
- [19] M. Jamei, A. Nisnevich, E. Wetchler, S. Sudat, E. Liu, Predicting All-Cause Risk of 30-Day Hospital Readmission Using Artificial Neural Networks, *PloS oOne* 12 (7) (2017) e0181173.
- [20] J. Futoma, J. Morris, J. Lucas, A Comparison of Models for Predicting Early Hospital Readmissions, *Journal of Biomedical Informatics* 56 (2015) 229–238.
- [21] L. Tong, C. Erdmann, M. Daldalian, J. Li, T. Esposito, Comparison of Predictive Modeling Approaches for 30-Day All-Cause Non-Elective Readmission Risk, *BMC Medical Research Methodology* 16 (1) (2016) 26.

- [22] S. B. Golas, T. Shibahara, S. Agboola, H. Otaki, J. Sato, T. Nakae, T. Hisamitsu, G. Kojima, J. Felsted, S. Kakarmath, et al., A Machine Learning Model to Predict the Risk of 30-Day Readmissions in Patients with Heart Failure: A Retrospective Analysis of Electronic Medical Records Data, *BMC Medical Informatics and Decision Making* 18 (1) (2018) 44.
- [23] A. Ashfaq, A. Sant'Anna, M. Lingman, S. Nowaczyk, Readmission Prediction Using Deep Learning on Electronic Health Records, *Journal of Biomedical Informatics* 97 (2019) 103256.
- [24] J. Davis, M. Goadrich, The Relationship Between Precision-Recall and ROC Curves, in: *Proceedings of the 23rd International Conference on Machine Learning*, 2006, pp. 233–240.
- [25] D. G. Kleinbaum, K. Dietz, M. Gail, M. Klein, M. Klein, *Logistic Regression*, Springer, 2002.
- [26] L. Rokach, O. Z. Maimon, *Data Mining with Decision Trees: Theory and Applications*, Vol. 69, World scientific, 2008.
- [27] L. Breiman, Random Forests, *Machine Learning* 45 (1) (2001) 5–32.
- [28] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu, LightGBM: A Highly Efficient Gradient Boosting Decision Tree, in: *Advances in Neural Information Processing Systems*, 2017, pp. 3146–3154.
- [29] V. Klema, A. Laub, The Singular Value Decomposition: Its Computation and Some Applications, *IEEE Transactions on Automatic Control* 25 (2) (1980) 164–176.

**Acknowledgement:** The work of Jerzy Krawczuk and Tomasz Łukaszk is partially supported by Białystok University of Technology [WZ/WI-IIT/3/2020]. The authors would like to acknowledge OwnedOutcomes Ltd. company for providing the data.

#### ***Przewidywanie ryzyka readmisji pacjentów z niewydolnością serca na podstawie danych z claim'ów***

**Streszczenie:** Wraz ze wzrostem wykorzystania rejestrów danych medycznych, eksploracja danych zyskała możliwość poprawy planowania leczenia i przyczynienia się do większej przeżywalności pacjentów. Techniki uczenia maszynowego stają się coraz bardziej popularne w szacowaniu prawdopodobieństwa readmisji szpitalnych. Właściwe wskazanie pacjentów wysokiego ryzyka może przyczynić się do zmniejszenia poziomu readmisji. Celem niniejszej pracy jest pokazanie, w jaki sposób można przeprowadzić estymację ryzyka readmisji i jakich wyników można się spodziewać dla dużej liczby epizodów niewydolności serca. Problem został przeanalizowany w badaniu empirycznym z wykorzystaniem czterech klasyfikatorów i kilku metod kodowania kodów diagnostycznych. Wyniki pokazują, że wydajność metod uczenia maszynowego mierzona polem pod krzywą ROC-AUC (*receiver operating characteristic curve*) dla readmisji z powodu niewydolności serca może wynosić nawet 0,675. Wyniki zostały uzyskane na bardzo dużej populacji (ponad 600 tys. epizodów opieki) z wykorzystaniem wyłącznie danych dostępnych na podstawie zgłoszeń medycznych, tzw. claim'ów, które zawierają głównie kody diagnoz w standardzie *International Statistical Classification of Diseases and Related Health Problems* w wersji 10 (ICD-10). Badanie pokazuje, że połączenie informacji o historii pacjenta z danymi z bieżącego epizodu może poprawić wyniki o ponad 3%. Klasyfikator LightGBM może być trenowany bardzo szybko na tak dużych zbiorach danych i zapewnia najlepsze wyniki. Wyniki te są nieco lepsze niż regresja logistyczna (tylko o około 0,7%), ale czas uczenia jest 10 razy krótszy.

**Słowa kluczowe:** *readmisja szpitalna, niewydolność serca, predykcja ryzyka, uczenie maszynowe, pole pod krzywą ROC, LightGBM, las losowy, regresja logistyczna*



# Rozdział IV

## Alternatywny algorytm dla transformaty Hough'a wyszukujący wzorce liniowe

Paweł Zabielski

Wydział Informatyki, Politechnika Białostocka

**Streszczenie:** Do dnia dzisiejszego, w celu rozpoznawania wzorców na różnego rodzaju obrazach czy grafikach, nadal wykorzystuje się metodę zaproponowaną w 1962 roku przez Paula Hough'a. Pierwsza wersja przedstawionego algorytmu używana była do wydobywania zależności liniowych, jednak została ona rozszerzona na ogólne klasy krzywych. Pomimo popularności tego rozwiązania posiada ono swoje wady. Do najważniejszych można zaliczyć jego złożoność pamięciową, jak i obliczeniową. W pracy tej została przedstawiona alternatywna metoda dla transformaty Hough'a. Pozwala ona na przeszukiwanie przestrzeni parametrów bez konieczności budowania akumulatorów.

**Słowa kluczowe:** transformata Hough'a, wzorce liniowe, przestrzeń cech, przestrzeń parametrów, funkcja kryterialna

## 1. Wprowadzenie

Techniki wykrywania wiedzy w dzisiejszych czasach są wykorzystywane praktycznie w każdej dziedzinie życia. Wiele z nich stosowanych jest podczas analizy obrazów. Algorytmy eksploracji danych mogą być używane do wydobywania informacji z różnego rodzaju ilustracji. Jednym z elementów, jaki może być rozważany jest m.in. zagadnienie wydobywania wzorców z obrazów dwuwymiarowych (2D) [1].

W rozdziale tym analizowany jest właśnie problem wydobywania wzorców liniowych z obrazów 2D. Do rozważenia są również wzorce bliskie wzorcom liniowym – wzorce kolinearne. Zastosowana została tu metoda minimalizacji wypukłych i odcinkowo-liniowych funkcji kryterialnych (metoda CPL) [2] i poprzez odpowiednie dostosowanie jej weryfikowane są jej możliwości wydobywania wzorców liniowych w obrazach. Niezbędnym elementem było właściwe przygotowanie danych, które zostały wykorzystane w obliczeniach. Wejściowy obraz został odpowiednio przekształcony. Do wykrywania krawędzi został użyty algorytm Canny'ego [3]. Przekształcone dane stanowiły odpowiednie wejście do dalszych obliczeń. Zastosowany algorytm wymiany rozwiązań bazowych w sposób efektywny był w stanie wyznaczać minimum



wypukłych i odcinkowo-liniowych funkcji kryterialnych. Algorytm ten podobny jest do algorytmu *Simplex*, używanego w programowaniu liniowym. W każdym kolejnym kroku następuje przejście do nowego wierzchołka.

Omawiana tu metoda docelowo przenoszona będzie na innego typu dane, jednak w monografii tej przedstawiona jest jako metoda służąca do wykrywania wzorców w obrazach. Jest to zupełnie nowy sposób i alternatywa dla transformacji Hough'a, która została przedstawiona i opatentowana w 1962 roku przez Paula Hough'a. Jednym z głównych zadań transformacji Hough'a jest właśnie znajdowanie zależności liniowych na obrazach [4]. Metoda ta jest do dnia dzisiejszego wykorzystywana. Dodatkowo została ona odpowiednio dostosowana do ogólnych klas krzywych [5].

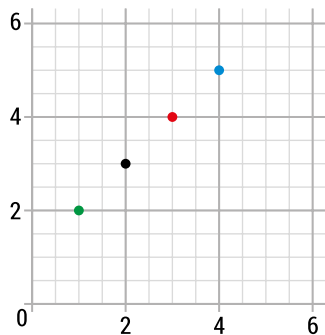
## 2. Główne zależności

Główną zależnością, która jest wykorzystywana zarówno w transformacji Hough'a, jak i w alternatywnym sposobie przedstawionym w pracy jest fakt, iż punkty położone współliniowo na płaszczyźnie w przestrzeni cech można opisać równaniem:

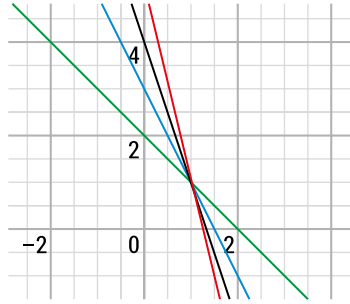
$$y = ax + b, \quad (1)$$

gdzie  $a$  jest nachyleniem (tanges kąta), zaś  $b$  wartością funkcji w punkcie 0. Odwzorowując dowolny punkt  $(x, y)$  znajdujący się w przestrzeni cech do przestrzeni parametrów  $a$  oraz  $b$  otrzymujemy prostą. Dodatkowo wszystkie proste w przestrzeni parametrów, które powstały poprzez odwzorowanie punktów leżących na prostej w przestrzeni cech, przecinają się dokładnie w jednym punkcie w przestrzeni parametrów [6]. Właśnie ta prawidłowość pozwala na zastosowanie odpowiednich metod do odnajdywania płaskich wzorców.

Dla lepszego zobrazowania omawianej wyżej zależności na rysunkach 1 i 2 zostały przedstawione relacje między punktami i prostymi w przestrzeni cech oraz parametrów:



RYSUNEK IV.1. Przedstawienie punktów w przestrzeni cech



RYSUNEK IV.2. Przedstawienie odwzorowania punktów z przestrzeni cech z Rys. IV.1 do przestrzeni parametrów

Zaznaczyć jednak trzeba, iż reprezentacja prostej poprzez postać kierunkową prostej nie jest wolna od wad. Problemy m.in. związane są z bardzo dużymi wartościami parametru  $a$  dla prawie pionowych linii oraz brakiem reprezentacji dla linii pionowych.

Podejściem, które niweluje przedstawione wyżej problemy jest parametryzacja w biegunowym układzie współrzędnych:

$$x \cos(\theta) + y \sin(\theta) = \rho, \tag{2}$$

gdzie  $\theta$  jest kątem nachylenia, zaś  $\rho$  odległością prostej od początku układu współrzędnych.

Propozycja metody przedstawionej w rozdziale wynika z faktu, iż algorytm transformacji Hough'a posiada wady. Główne problemy są związane ze złożonością obliczeniową oraz pamięciową. Wynikają one z konieczności przygotowywania akumulatorów, na których wykonywane są obliczenia. Powoduje to wykładniczy wzrost złożoności obliczeń wraz ze wzrostem wielkości wymiarów danych wejściowych [4].

W prezentowanej pracy zostanie przedstawiona alternatywna metoda efektywnego wyszukiwania wzorców liniowych w obrazach dwuwymiarowych.

### 3. Wierzchołki w przestrzeni cech

Do wykonania obliczeń niezbędne było odpowiednie przygotowanie danych wejściowych. Obraz wejściowy poddawany był obróbce, aby docelowo przedstawić go w postaci zbioru danych, który składał się z dwuwymiarowych wektorów cech  $\mathbf{x}_j = [x_{j1}, \dots, x_{jn}]^T$  ( $\mathbf{x}_j \neq \mathbf{0}$ ), gdzie  $j = 2$ . W taki sposób zbudowana została macierz o  $m$  wierszach i 2 kolumnach. Każdy z wierszy przedstawiony był jako współrzędne  $x$  oraz  $y$ . Taka reprezentacja umożliwia przedstawienie w dwuwymiarowej przestrzeni cech

$F[n]$  ( $\mathbf{x}_j \in F[n]$ ) obrazu wejściowego jako punkty. Ogólnie zakłada się, że odpowiednie wartości poszczególnych cech to liczby rzeczywiste  $x_{ji}$ . Rozważany zbiór danych  $C$  składał się z  $m$  wektorów cech:

$$C = \{\mathbf{x}_j\}, \text{ gdzie } j = 1, \dots, m. \quad (3)$$

Płaszczyznę wierzchołkową  $P_k(\mathbf{x}_{j(1)}, \dots, \mathbf{x}_{j(l+1)})$  o wymiarze  $(l - 1)$  w dwuwymiarowej przestrzeni cech  $F[n]$  zdefiniujemy za pomocą  $l$  liniowo niezależnych wektorów cech  $\mathbf{x}_{j(i)}$ , zgodnie z [6]:

$$P_k(\mathbf{x}_{j(1)}, \dots, \mathbf{x}_{j(l)}) = \{\mathbf{x}: \mathbf{x} = \alpha_1 \mathbf{x}_{j(1)} + \dots + \alpha_l \mathbf{x}_{j(l)}\}. \quad (4)$$

Parametry  $\alpha_i$  powinny spełniać warunek  $\alpha_1 + \dots + \alpha_l = 1$  [7].

## 4. Wierzchołki w przestrzeni parametrów

Wektory cech  $\mathbf{x}_j$  ze zbioru  $C$  pozwalają na zdefiniowanie następujących hiperpłaszczyzn  $h_j$  w przestrzeni parametrów:

$$(\forall \mathbf{x}_j \in C) h_j = \{\mathbf{w}: (\mathbf{x}_j)^T \mathbf{w} = 1\}, \quad (5)$$

gdzie  $\mathbf{w}_k = [w_{k,1}, \dots, w_{k,n}]^T \in R^n$  jest wektorem wag.

Każdy z kolejnych wierzchołków może być opisany jako przecięcie pewnej ilości hiperpłaszczyzn  $h_j$ . W związku z tym wierzchołek  $\mathbf{w}_k$  może być zdefiniowany następująco:

$$(\forall j \in J_k) \mathbf{w}_k^T \mathbf{x}_j = 1. \quad (6)$$

Wierzchołek jest wierzchołkiem zdegenerowanym, gdy jest punktem przecięcia odpowiednich ilości hiperpłaszczyzn  $h_j$ ; są to właśnie punkty przecięcia w przestrzeni parametrów. Docelowo zagadnienie związane jest ze znalezieniem najbardziej zdegenerowanych wierzchołków – każdy krok algorytmu powinien powodować przejście do kolejnego wierzchołka zdegenerowanego. W taki sposób aktualizowany jest szukany wektor wag.

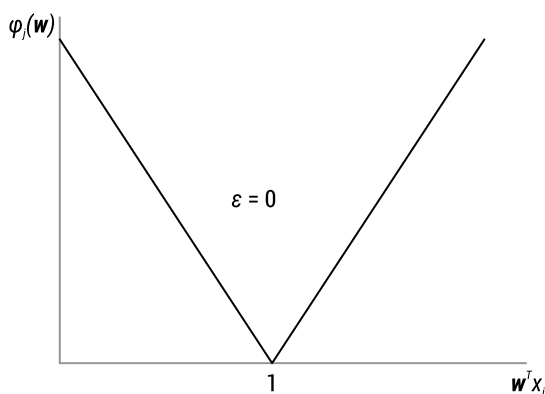
W publikacji tej przedstawione jest to rozwiązanie oparte na minimalizacji wypukłych i odcinkowo-liniowych (typu *CPL*) funkcji kryterialnych [7].

## 5. Funkcja kryterialna

Wektory cech  $\mathbf{x}_j$  z zbioru  $C$  pozwalają na zdefiniowanie następującej funkcji kary:

$$(\forall \mathbf{x}_j \in C(1)) \quad \varphi_j(\mathbf{w}) = |1 - \mathbf{x}_j^T \mathbf{w}|. \quad (7)$$

Jest to funkcja wypukła i odcinkowo-liniowa (typu *CPL*) [7]. Rysunek IV.3 przedstawia postać funkcji kary:



RYSUNEK IV.3. Funkcja kary

Funkcja kryterialna  $\Phi(\mathbf{w})$  jest zdefiniowana jako suma funkcji kar  $\varphi_j(\mathbf{w})$  (7), która jest zależna od  $m$  wektorów cech  $\mathbf{x}_j$ :

$$\Phi(\mathbf{w}) = \sum \varphi_j(\mathbf{w}) \quad j=1, \dots, m. \quad (8)$$

Można wykazać, że minimalna wartość  $\Phi(\mathbf{w})^*$  wypukłej i odcinkowoliniowej (typu *CPL*) funkcji kryterialnej  $\Phi(\mathbf{w})$  (8) znajduje się w jednym z wierzchołków  $\mathbf{w}_k$  [8]:

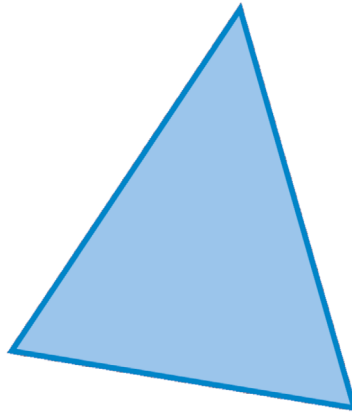
$$(\exists \mathbf{w}_k^*) (\forall \mathbf{w}) \Phi(\mathbf{w}) \geq \Phi(\mathbf{w}_k^*) = \Phi_k^* \geq 0. \quad (9)$$

Przedstawiony algorytm wymiany rozwiązań bazowych ma za zadanie w sposób efektywny znaleźć optymalny wierzchołek  $\mathbf{w}_k^*$ , w którym wartość funkcji kryterialnej jest minimalna.

## 6. Kolejne kroki wykrywania zależności liniowych

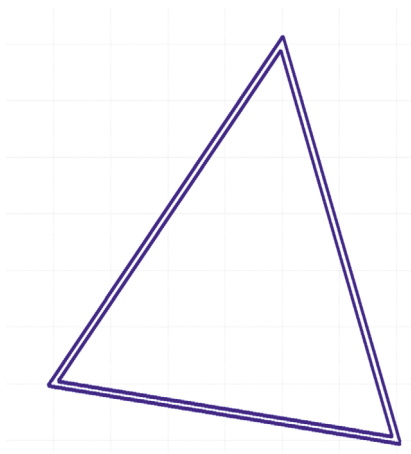
Pierwszym etapem niezbędnym do przeprowadzenia obliczeń jest odpowiednie przygotowanie danych wejściowych. Jednym z ważnych w praktyce zagadnień jest tu problem redukcji zakłóceń (szumów) w obrazach dwuwymiarowych [9].

Przykładowym obrazem, który został wykorzystany do obliczeń, jest dwuwymiarowy obraz znajdujący się na rysunku IV.4, przedstawiający pewien kształt (Źródło: Triangle illustration – Trójkąt – Wikipedia, Wolna encyklopedia, rok: 2020):



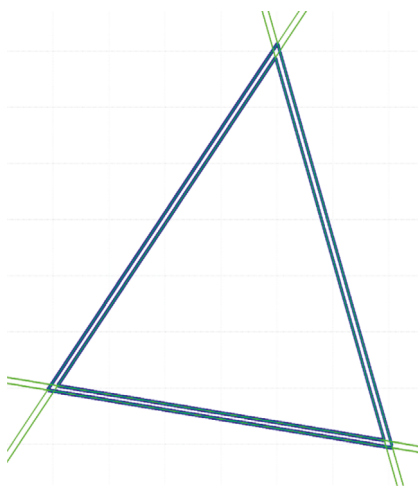
RYSUNEK IV.4. Jeden z dwuwymiarowych obrazów, na których testowany był algorytm mający na celu wykrywanie wzorców liniowych

To właśnie taki obraz na wejściu przekształcany jest do postaci, która może być wykorzystana przez algorytm. Przykładowa reprezentacja tego obrazu, która została wykorzystana do dalszych obliczeń jest przedstawiona na rysunku IV.5:



RYSUNEK IV.5. Reprezentacja obrazu, która zostaje wykorzystana do dalszych obliczeń

Poprzez minimalizację funkcji kryterialnej algorytm porusza się po wierzchołkach zdegenerowanych. Jeśli napotka mocno zdegenerowany wierzchołek – jest on traktowany jako jedno z rozwiązań i jako poszukiwana zależność liniowa. Analogicznie poprzez wyszukiwanie kolejnych wierzchołków zdegenerowanych wyszukiwane są pozostałe wzorce liniowe na danym obrazie. Rysunek IV.6 przedstawia wszystkie zależności liniowe, które algorytm jest w stanie znaleźć na podanym na wejściu obrazie dwuwymiarowym:



RYSUNEK IV.6. Znalezione rozwiązania – wykryte zależności liniowe

## 7. Wnioski

Niniejsza praca miała za zadanie przedstawienie alternatywnego algorytmu do transformaty Hough'a służącego m.in. do wyszukiwania wzorców liniowych. Metoda opiera się na zależnościach pomiędzy przestrzenią cech a parametrami oraz na minimalizacji funkcji kryterialnych typu *CPL*. W rozdziale pokazano, w jaki sposób metoda działa na obrazach dwuwymiarowych i jak wyglądają kroki doprowadzające do rozwiązań. Ważne jest to, iż w zaproponowanej metodzie, w przeciwieństwie do powszechnie stosowanego algorytmu transformacji Hough'a, nie są budowane duże akumulatory, które mają wpływ na złożoność pamięciową i obliczeniową.

Aktualnie będą następowały próby użycia algorytmu w innych typach zbiorów wejściowych. Docelowo planuje się użycie metody również w wielowymiarowych danych [10]. Miałoby to zastosowanie w przetwarzaniu danych genetycznych. W tym celu planuje się wykorzystanie danych wielowymiarowych rzeczywistych oraz ewentualne dostosowanie lub optymalizację aktualnej wersji algorytmu.

## Literatura

- [1] Duda, R. O., Hart, P. E. and Stork, D. G.: *Pattern Classification*, John Wiley & Sons, 2012.
- [2] Bobrowski L.: *Eksploracja danych oparta na wypukłych i odcinkowo-liniowych funkcjach kryterialnych*, Wydawnictwa Politechniki Białostockiej, Białystok 2005.
- [3] Canny, J. (1986). *A Computational Approach to Edge Detection*. IEEE Trans. PAMI, 8(6):679–698.
- [5] Hough P., *Method and Means for Recognizing Complex Patterns*, U. S. Patent 3,069,654, 1962.
- [6] Duda O. R., Hart P E., *Use of the Hough Transformation to Detect Lines and Curves in Pictures*, Communications of the ACM, vol.15, s. 11–15, 1972.
- [7] Illingworth J., Kittler J., *A Survey of the Hough Transform*, Computer Vision, Graphics and Image Processing, 44, 87–116, 1988.
- [8] Bobrowski, L.: *Discovering Main Vertexical Planes in a Multivariate Data Space by Using CPL Functions*, pp. 200–213 in: *ICDM 2014*, Ed. Perner P., Springer Verlag, Berlin, 2014.
- [9] Simonnard M.: *Linear Programming*, Prentice – Hall, New York, Englewood Cliffs, 1966.
- [10] Bobrowski, L., Zabielski P.: *Flat Patterns Extraction with Collinearity Models*, EUROSIM, Oulu, 2018, pp. 518–524.
- [11] Bobrowski L. Zabielski P.: *Feature (Gene) Clustering with Collinearity Models*, Computational Collective Intelligence: 13<sup>th</sup> International Conference, ICCCI 2021, Rhodes, Greece, September 29 – October 1, 2021, Proceedings / Wojtkiewicz Krystian [et al.] (eds.), Communications in Computer and Information Science, 2021, vol. 12876, Cham, Springer, pp.403–415, ISBN 978-3-030-88080-4.

### ***An alternative algorithm for the Hough transform that searches for linear patterns***

**Abstract:** To this day, the method proposed in 1962 by Paul Hough is still used to recognize patterns in various types of images or graphics. The first version of this algorithm was applied to extract linear patterns, but it was extended to general classes of curves. Despite the popularity of this solution, it has its drawbacks. The most significant drawback is its memory and computational complexity. This article presents an alternative method for the Hough transform that allows one to search the parameter space without building accumulators.

**Keywords:** *the Hough transform, linear patterns, feature space, parameter space, criterion function.*

# Rozdział V

## Image processing methods in semi-automatic mobile application testing framework

*Maciej Szymkowski, Tomasz Łukaszuk, Andrzej Chmielewski  
The Faculty of Computer Science,  
Białystok University of Technology, Białystok, Poland*

**Abstract:** Recently, the mobile application market has been growing. Each year one can observe many new programs for diversified mobile operating systems such as Android or iOS. One of the most critical problems regarding mobile application development is connected with testing. It was proven/proved that manual testing is ineffective, and sometimes, it cannot guarantee satisfactory results. The crucial aspects that have to be covered in such a procedure are screen resolutions, operating systems, versions of OS, and hardware configuration. A conceptual approach to a semi-automatic mobile application testing framework can be proposed to solve these issues. In this work, the Authors would like to describe precisely which image processing methods can be used in such a solution. Both segmentation and template matching methods for test reproduction on different devices were considered. The article ends with a presentation and discussion of the results of the conducted experiments.

**Keywords:** *image processing, mobile applications, testing framework, segmentation, template matching, k-Means*

## 1. Introduction

On the basis of the statistics shown in [1], in 2017, more than 1.5 billion mobile applications were registered in diversified mobile stores (such as Google Play or Apple Store). Moreover, in 2019, users downloaded at least 204 billion applications into their mobile phones. The statistics show that the mobile application market is still growing, and the number of developed applications breaks the record every year.

One of the essential stages connected with mobile application development is testing. Detecting errors, mistakes, or even backdoors is fundamental and has to be done before an application is made available for end users. It has to be claimed that the testing team should carefully check the backend part of an application (with all developed solutions) and its visual mechanisms (especially the availability of different clickable elements). This task is crucial to properly develop mobile applications



and prepare them for the final users. However, it is challenging to manually check how an application deals with diversified parameters of mobile devices such as screen resolutions, operating system and its version, and hardware configuration (even though there are some worked-out standards).

To deal with this problem, the authors of this paper would like to propose their semi-automated mobile application testing framework. However, it has to be claimed that there are no such solutions on the market. The main aim of the proposed program is to record the test and then repeat it at different screen resolutions (in the part connected with the visual side of the application). In this paper, the authors present detailed descriptions connected with mechanisms responsible for image processing. In the first stage (based on different image segmentation techniques, including techniques based on machine learning), detailed information about controls available on the screen is obtained. In the second step, template matching techniques are used to study whether the test can be run properly at different screen resolutions. It will allow one to get accurate information on whether the application is presented in the same manner in analysed environments (i.a., mobile phone, tablet, or iPad).

This work is organised as follows: in the first section, different approaches to mobile testing applications and image processing methods are described. In the second one, the proposed approach is described and shortly discussed. The third section is connected with the results of the conducted experiments. Each of them is also described, and the results are given. Finally, the conclusions and information about future work are presented.

## 2. Known approaches

One can find multiple diversified approaches regarding image recognition or template matching in the literature. These subjects were described in diversified works and papers. On the other hand, the image segmentation process is also well-known and was precisely described in the literature. One novelty that different researchers have not yet profoundly analysed is using these algorithms for image testing purposes. In this chapter of the following work, the analysis of the current state of the art is presented. To be precise, each particular algorithm is described in a separate subsection. Three main parts of the literature review have been observed: algorithms connected with segmentation, ideas regarding template matching, and recently published papers describing automatic (or semi-automatic) ideas for mobile application testing.

### 2.1. Segmentation

In the literature, one can find different ideas connected with image segmentation. The main aim of this operation is to deal with separating foreground objects from the image's background. Segmentation has to be used in diversified ideas

and algorithms. On the one side, it can be used in biometric algorithms (e.g., to extract fingerprint area from the image), while it can also be helpful in object detection. In this section, some approaches to image segmentation will be presented.

One of the fascinating approaches was presented in [2]. In this paper, the Authors proposed a clustering-based segmentation algorithm. It consists of two main stages; the first is pre-segmentation with mean-shift algorithm. Based on this operation, the potential objects are marked. Then, the proper areas are selected in the second step, which is done with the Hierarchical Clustering approach. The idea is fascinating because it expands the idea of the k-Means-based segmentation [3] (k-Means is one of the most well-known approaches to clustering).

A completely different idea was presented in [4]. In this case, the Authors used the Firefly algorithm to precisely split the image into objects and background. As it was mentioned in the paper, the basic version of the Firefly approach is not effective enough in the case of multiple different objects visible in the image. It is connected with the fact that it can be stopped on some specific local optima. The Authors provided specialised strategies; for instance, diversity enhancing strategy with the Cauchy mutation, to escape the local optima and accelerate the convergence. As was shown in the paper, the proposed approach can efficiently segment even multilevel images.

An intriguing algorithm was also proposed in [5]. The approach is based on a typical idea connected with image binarization. The specific threshold is applied to detect both object and background. Interestingly, such an approach is chosen in case of an image corrupted by Gaussian noise. The results have shown that it is possible to detect foreground and background in the image, even when some noise is observable.

An artificial intelligence-based idea was proposed in [6]. In the paper, the Authors stated that most of the recently used algorithms for segmentation are supervised. It means that accurate information about masks, images, objects, and backgrounds must be provided at the beginning of the algorithm learning. The Authors described their unsupervised approach to the segmentation problem. The idea relies on the fact that it can be possible to change colours or textures without changing the overall distribution of the dataset. Moreover, they used adversarial architecture; the object-candidate mask is marked, and then the object is redrawn. The work showed that the proposed approach can guarantee satisfactory results, and its precision is also acceptable.

One can also easily find diversified review approaches to recent ideas connected with segmentation techniques. The most intriguing ones were described and presented in [7–9].

## 2.2. Template matching

Template matching is one of the approaches to the fundamental image processing problem, namely locating the region of interest [10]. It is a technique for finding small

parts of a source image which match a template image. Template matching methods can be classified into two basic categories: template-based approach and feature-based approach [11]. Both categories of methods have advantages and disadvantages.

In the template-based approach, also known as the area-based approach, we operate directly on pixel values. The template image is moved over the source image to enable the calculation of a specific numerical measure of similarity to the patch overlaid. Template matching techniques like normalized cross-correlation, cross-correlation, and the sum of squared differences are applied here [12]. Template-based methods are relatively easy to implement and work very well, even in cases where numerous and substantial image features cannot be determined in the template image. However, their drawbacks include low robustness to scaling and rotation of the template or source images. More effective algorithms have been proposed to overcome these difficulties [13]. These have come at the cost of increasing source image pattern finding complexity.

The feature-based approach involves extracting from images their features, such as fixed shapes, colours, and textures, and then matching the corresponding features on the template image and the source image. The detected features represent specific semantic structures in the image. They can be divided into corner features, blob features, line/edge and morphological region features, but the most popular and useful features are the so-called *interest points* or *keypoints*. Besides the mentioned hand-crafted features, the most intensively currently developed branch is learning-based feature detectors. In this area, neural networks and deep learning models are used with success. In the feature-based template matching approach, in addition to feature detection, we distinguish the stages of creating feature descriptors and matching corresponding keypoints and descriptors from template and source images. A very valuable and detailed overview of the current state of the art in feature-based image matching can be found in [14]. Most of the issues presented in this work also apply to feature-based template matching. For instance, the feature-based approach handles scaling, rotation, brightness changes, or blurring of images relatively well. However, it may perform worse when images have a small number of features, or different fragments (objects) have the same features.

### 2.3. Other testing approaches

Diversified approaches to testing applications and algorithms are also observable in the literature. However, most of them are connected with manual procedures. Moreover, only a small amount of recently published works is related to testing in an automated manner (sometimes these papers present even only conceptual ideas instead of implemented algorithms). Therefore, the authors of this work briefly describe selected approaches to mobile application testing in this subsection.

The first paper [15] provides an engaging analysis of the challenges and tools for mobile application testing. The Authors present a comparison between diversified

testing types. For instance, they mention security tests, performance tests, and functional tests. It means that the researchers only analyse high-level testing. In the paper, one cannot read any information connected with unit testing or other code testing procedures. Moreover, the Authors compare different tools for mobile application testing. For instance, Appium, Selendroid, or Monkeyrunner are indicated. This work is fascinating and presents broad knowledge connected with mobile application testing.

A different perspective on mobile application testing was presented in the paper [16]. The Authors analysed diversified tools connected with software engineering. For instance, bug and error reporting tools were mentioned. In this case, the work is interesting; however, it mainly applies traditional tools & software engineering approaches to mobile application development. The most interesting part of the work is comparing diversified tools for application testing. Furthermore, it shows which programs can be directly used for testing mobile applications.

One can also find approaches to testing specific parameters connected with mobile applications. For instance, safety testing with penetration procedures was described in [17], while performance testing was shown in the paper [18].

It also has to be claimed that mobile application testing is mainly carried out with emulators. It means that the mobile application is run with computer resources, not on the smartphone. This way is much easier and more efficient because smartphones mostly have lower quality components (CPU or RAM) than stationary computers or laptops. However, in the literature one can also observe an interesting approach [19] where the whole testing procedure was carried out on a smartphone with its resources. The Authors claimed that their idea provided efficient and accurate results even with low-budget smartphones. However, the main question is connected with how the tests were performed. It is unclear how the tests written for traditional (or web) application testing were converted to be run with a microcontroller placed in the smartphone. It has to be claimed that microcontrollers used in mobile devices (as ARM) have different architectures than those used in stationary computers or laptops.

The literature reports a general lack of automated testing adoption among mobile developers in the industry, especially regarding graphical interfaces. Some related aspects are presented in [20], where important players in the mobile application market share their experiences in this field, trying to automate this process to the greatest possible extent.

### 3. Proposed approach

This chapter, will present the general concept of the mobile application testing system and information about segmentation and template matching. Some arguments about why these two operations were used and what benefits they can guarantee in the case of the proposed framework will be mentioned. Moreover, while working under these

steps, there have been benefits from the previous experiences connected with image processing and testing [21, 22].

### 3.1. The main idea of the system

The proposed framework has to deal with diversified parameters connected with mobile applications and devices. These attributes are screen resolution, operating system and its version and hardware configuration. A visual approach to this problem is proposed. The system will consist of three main stages. Each of them is presented in Fig. V.1.

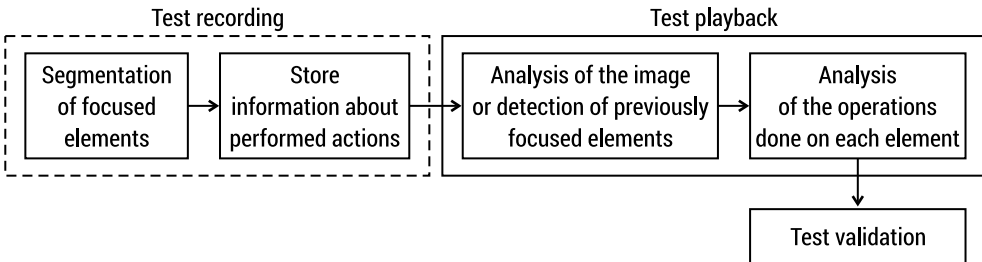


FIGURE V.1. Block scheme of the proposed approach

In the first step, we would like to record the test. It means that the user has to move the application forward manually. His every action will cause some specific procedures to be done. Each focused element will be segmented – we will save information about the position, what kind of control was pointed, and what action was done on this object. Moreover, the proposed framework will store information about the transition between different application screens. It will allow one to precisely store what kind of actions were taken and what transitions were made.

In the second stage, the scenario on different mobile devices will be reproduced (as stated before, it is related to different parameters). The stored scenario will be passed step-by-step. It means that we will analyse whether previously segmented elements are available in the image (it will be done with template matching operation). If a proper object is found, one can move on to take proper action on it (for instance, click and enter the text). In this stage, as an input, a current application screenshot and the scenario recorded in the previous stage will be used.

The final step of the system is test validation. All steps are carefully examined to see if they were followed. It means that the first situation in which an element was not found in the image ends the test with failure. In the other case, if all tests are passed, one can claim that the test was finished successfully.

The sample scenario was presented in Fig. V.2., pointing out what elements were clicked and what kind of transitions were made. The scenario in the application used is not saved, as presented in Fig. V.2. Additional files in which information about focused elements and related actions are saved are used.

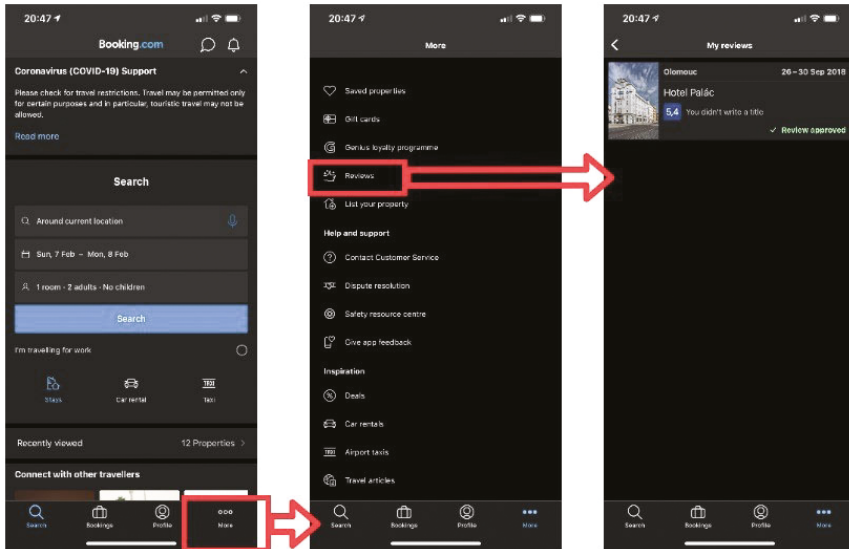


FIGURE V.2. Sample scenario of testing an application – booking.com. Each segmented element is marked with a red square. After clicking on the selected element, the arrow will show the result

### 3.2. Segmentation

Segmentation plays an essential role in the proposed approach. It is connected with the fact that by this operation one can check which control the user acted (for instance, click, data input or value change). The information gained in this stage is also helpful in the further step, namely template matching. Moreover, it has to be claimed that the result of this stage is a file in which all extracted elements are shown with additional information about their position in the image (expressed in the form of percentage – it can be useful in the case of different screen resolutions). During experiments, diversified approaches connected with segmentation (for instance, those described in section 2.1). However, due to the complexity of different applications (multiple different controls and styles) it was impossible to prepare one simple algorithm for each of them.

In order to deal with this problem, the image’s complexity is measured. First, it is done on the basics of simple statistical methods such as variance and standard deviation. Then the proper method is selected to split the image. If the variance is low, one uses an approach based on the thresholding technique, although when the statistical value is high, then Machine Learning-based approach is run (based on the k-Means clustering algorithm). The results obtained with these methods are presented in Fig. V.3 and Fig. V.4 respectively.

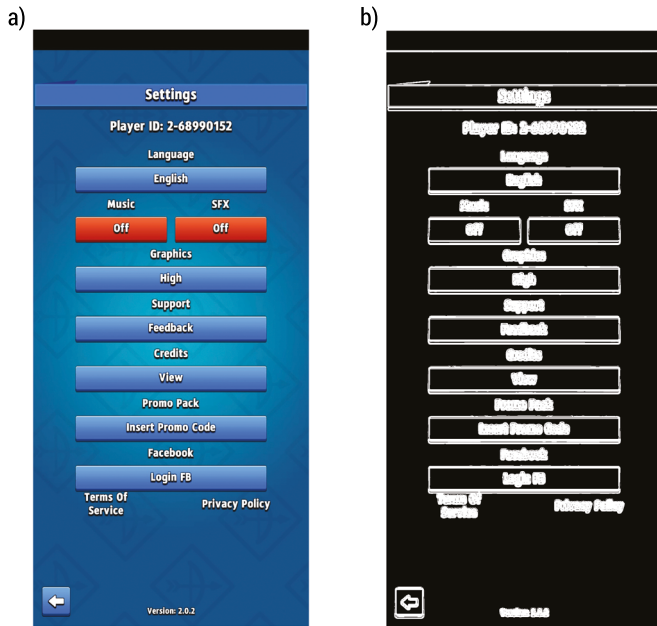


FIGURE V.3. Simple segmentation operations – original sample (a) and processed image (b)

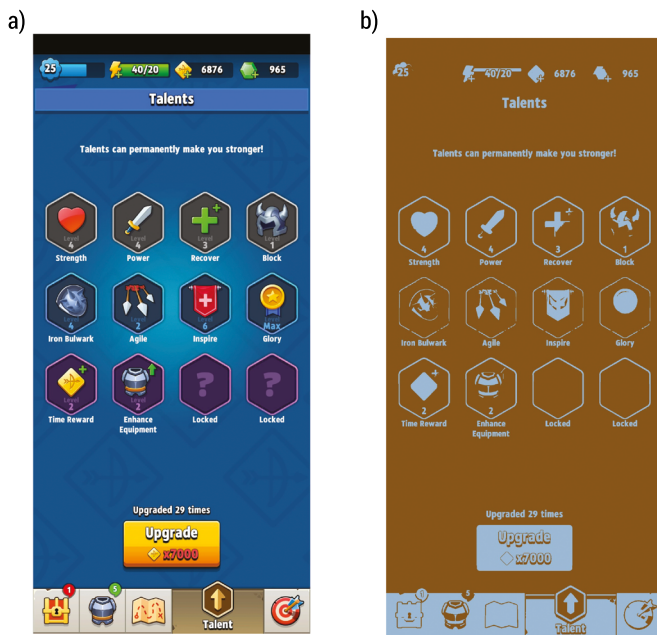


FIGURE V.4. Segmentation with k-Means – original image (a) and processed sample (b)

### 3.3. Template matching

The specifics of the application of template matching methods mean that one is not interested in robustness to rotation; only robustness to image scaling is essential. When playing the test scenario, one may have to deal with an image at a different resolution than the one from which the template was cut (the image will not be rotated). Furthermore, the elements (template images) to be searched for may vary in complexity, with more or less detail, from simple single-colour geometric figures to graphically complex images. Given these requirements, it was decided to use an ensemble model consisting of a template-based algorithm and a feature-based algorithm, which independently perform the matching, and the results generated by them are passed to the input of the voting module that produces the final answer (see Fig. V.5).

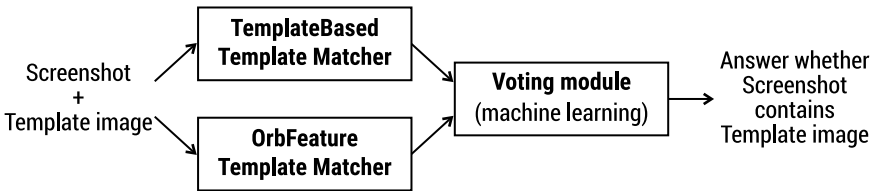


FIGURE V.5. Scheme of the template matching algorithm used in the proposed framework

The template-based algorithm calculates the normalised correlation coefficient [23] between the template image and the tested screenshot fragment. The highest coefficient value obtained is returned as the evaluation result. The feature-based algorithm is derived from the modified ORB algorithm [24]. The voting module, on the other hand, is the random forest classifier [25].

## 4. Experiments

In this section, the authors briefly describe the conducted experiments. This section is divided into the one in which segmentation is described and the other for template matching.

### 4.1. Segmentation

While working under the segmentation process, the Authors tested multiple different approaches. Beginning from classical ones such as watershed algorithm [26] and those more complex as described in section 2.1. The first idea connected with the segmentation was to work out one specialized algorithm for all possible images. In the first iteration, the Authors tried to use histogram-based approaches. The method used



a global histogram for images with low complexity, while local histograms were analysed for those with huge complexity. However, in the further process it was observed that the method is inefficient and the results are not satisfactory. To deal with this problem, the Authors moved to Machine Learning-based algorithms as segmentation with k-Means. However, it was observed that it can guarantee efficient and satisfactory results in the case of complex images, but in the case of simple ones, it is not efficient enough. For this aim, the Authors prepared a simple algorithm based on image processing methods for low-complexity images. Its block diagram is presented in Fig. V.6.

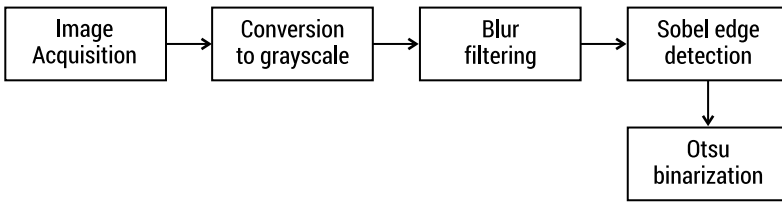


FIGURE V.6. Block diagram of the algorithm for low complexity images

It also has to be claimed that the approach proposed in this paper, based on statistical measures (by which images are classified as being either of low or high complexity), guaranteed the best, most precise results.

## 4.2. Template matching

Numerous experiments have confirmed that the OrbFeature Template Matcher algorithm (module) performs better with more complex templates, while the TemplateBased Template Matcher algorithm is better with simpler ones. The combination algorithms' advantages and the power of machine learning resulted in very high performance for all types of template images. The classification accuracy at the current stage of development, depending on the mobile application tested, was 85% and higher.

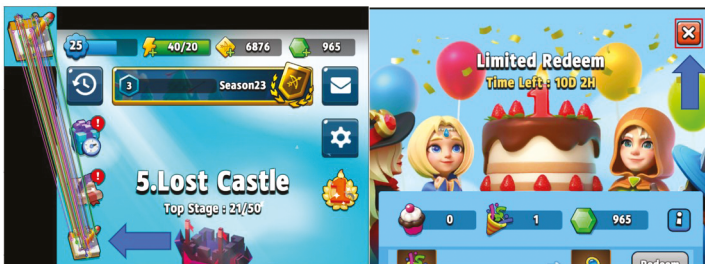


FIGURE V.7. Examples of elements (template images) which are better handled by the feature-based algorithm (left) and which are better handled by the template-based algorithm (right)

## 5. Conclusions and future work

Mobile games and applications are one of the most popular solutions nowadays. Their vast popularity is connected with the fact that one can use them in his mobile device without any additional modules. Only a simple smartphone is needed to run these applications.

However, one of the most important problems connected with mobile applications is how they can be tested. Some procedures described in the literature were worked out; however, they include only manual testing without specialised automatic approaches. Due to that, the Authors proposed their idea regarding semi-automatic mobile application testing. Such an approach allows one to precisely test mobile applications from the visual side (even with different screen resolutions, operating systems, their versions, and hardware configuration). Furthermore, the most critical stages of image processing, namely segmentation and template matching, were described in this paper.

All algorithms were implemented using Python and OpenCV framework in a real development environment. The results have shown that both used segmentation and template matching algorithms are efficient enough and can guarantee precise results in object extraction and comparison.

The authors' current work is to prepare the implementation of the whole system. Moreover, some specialised artificial intelligence-based algorithms that can be used directly in the system are still being created. In the nearest future, the authors would also like to share all sample databases. Unfortunately, right now, there are no such bases available on the Internet.

## References

- [1] <https://websitebuilder.org/blog/app-usage-statistics/> (Accessed: 07.02.2021)
- [2] Tian, L., Han, L., Yue, J.: "Research on Image Segmentation based on Clustering Algorithm", *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 9, no. 2, 2016, pp. 1–12.
- [3] Dhanachandra, N., Mangleam, K., Jina Chanu, Y.: "Image Segmentation Using k-Means Clustering Algorithm and Subtractive Clustering Algorithm", *Procedia Computer Science*, vol. 54, 2015, pp. 764–771.
- [4] Chen, K., Zhou, Y., Zhang, Z., Dai, M., Chao, Y., Shi, J.: "Multilevel Image Segmentation Based on an Improved Firefly Algorithm", *Mathematical Problems in Engineering*, vol. 2016, 2016, pp. 1–12, DOI: 10.1155/2016/1578056.
- [5] Abubakar, F.: "Study of Image Segmentation Using Thresholding Technique on a Noisy Image", *International Journal of Science and Research*, vol. 2, issue 1, 2013, pp. 49–51.
- [6] Chen, M., Artieres, T., Denoyer, L.: "Unsupervised Object Segmentation by Redrawing", *33rd Conference on Neural Information Processing Systems (NeurIPS)*, 2019, Vancouver, Canada, 2019, pp. 1–12.
- [7] Zaitoun, N., Aquel, M.: "Survey on Image Segmentation Techniques", *Procedia Computer Science*, vol. 65, 2015, pp. 797–806.
- [8] Khan, M.: "A Survey: Image Segmentation Techniques", *International Journal of Future Computer and Communication*, vol. 3, no. 2, 2014, pp. 89–93.

- [9] Noble, A., Boukerroui, D.: "Ultrasound Image Segmentation: A Survey", *IEEE Transactions on Medical Imaging*, vol. 25, issue 8, 2006, pp. 987–1010.
- [10] Brunelli, R.: "Template Matching Techniques in Computer Vision: Theory and Practice", John Wiley & Sons, 2009.
- [11] Babbar, G., Bajaj, P., Chawla, A., Gogna, M.: "Comparative Study of Image Matching Algorithms", *International Journal of Information Technology and Knowledge Management*, 2(2), 2010, pp. 337–339.
- [12] Hisham, M. B., et al.: "Template Matching Using Sum of Squared Difference and Normalized Cross Correlation" in: 2015 IEEE student conference on research and development (SCORED). IEEE, 2015. p. 100–104.
- [13] Lee, W.-C., Chen, C.-H.: "A Fast Template Matching Method for Rotation Invariance Using Two-Stage Process" in: 2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing. IEEE, 2009. p. 9–12.
- [14] Ma, J., Jiang, X., Fan, A., Jiang, J., Yan, J.: Image matching from handcrafted to deep features: A survey. *International Journal of Computer Vision*, 2020, pp. 1–57.
- [15] Ali, U., Arif, K. S.: "Mobile Application Testing Tools and their Challenges: A Comparative Study", *International Conference on Computing, Mathematics and Engineering Technologies, ICOMET 2019, Sukkur, Pakistan, Proceedings*, pp. 1–6.
- [16] Linares-Vasquez, M., Moran, K., Poshyvanik, D.: "Continuous, Evolutionary and Large-Scale: A New Perspective for Automated Mobile App Testing", arXiv: 1801.06267v1 [cs.SE], 2018.
- [17] Alanda, A., Satria, D., Mooduto, H., Kurniawan, B.: "Mobile Application Security Penetration Testing Based on OWASP", *Materials Science and Engineering*, vol. 846, 2020, pp. 1–12, DOI: 10.1088/1757-899X/846/1/012036.
- [18] Norberto, M., Gaedicke, L., Bernardino, M., Legramante, G., Basso, F. P., Macedo Rodrigues, E.: "Performance Testing in Mobile Application: A Systematic Literature Map", SBQS' 19: Proceedings of the XVIII Brazilian Symposium on Software Quality, 2019, pp. 99–108, DOI: 10.1145/3364641.3364653.
- [19] Nagowah, L., Sowamber, G.: "A Novel Approach of Automation Testing on Mobile Devices", 2012 International Conference on Computer & Information Science, ICCIS 2012, 12–14 June, 2012, Kuala Lumpur, Malaysia, Proceedings, DOI: 10.1109/ICCISci.2012.6297158
- [20] Coppola, R., Ardito, L., Morisio, M. and Torchiano, M.: "Mobile Testing: New Challenges and Perceived Difficulties From Developers of the Italian Industry," in *IT Professional*, vol. 22, no. 5, 2020, pp. 32–39.
- [21] Parfieniuk, M., Łukaszuk, T., Grześ, T.: "A Scalable Distributed Architecture for Emulating Unreliable Networks for Testing Multimedia Systems", Choras, R. (Eds.): "Image Processing and Communications Challenges 5", *Springer Advances in Intelligent Systems and Computing*, vol. 223, 2014, pp. 309–316.
- [22] Szymkowski, M., Saeed, E., Omieljanowicz, M., Omieljanowicz, A., Saeed, K., Mariak, Z.: "A Novelty Approach to Retina Diagnosing Using Biometric Techniques with SVM and Clustering Algorithms", *IEEE Access*, vol. 8, 2020, pp. 125849–125862, DOI: 10.1109/ACCESS.2020.3007656.
- [23] Briechle, K., Hanebeck U.D.: "Template Matching Using Fast Normalized Cross Correlation" in: *Optical Pattern Recognition XII*. International Society for Optics and Photonics, 2001. p. 95–102.
- [24] Rublee, E., Rabaud, V., Konolige, K.: "ORB: An Efficient Alternative to SIFT or SURF", *Proc of IEEE International Conference on Computer Vision*, 2011, pp. 2564–2571.
- [25] Breiman, L.: "Random Forests", *Machine learning* 45.1, 2001, pp. 5–32.

- [26] [https://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_imgproc/py\\_watershed/py\\_watershed.html](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_watershed/py_watershed.html) (Accessed: 09.02.2021).

**Acknowledgement:** This work was supported by the grant POIR.01.01.01-00-1298/19 from GameLiveOps Ltd. and funded with financial resources by The National Center for Research and Development (NCBiR) in Poland.

***Metody przetwarzania obrazów we framework’u do półautomatycznego testowania aplikacji mobilnych***

**Streszczenie:** W ostatnim czasie rynek aplikacji mobilnych rośnie w siłę. Każdego roku powstaje ogromna liczba nowych programów na różne mobilne systemy operacyjne, takie jak Android czy iOS. Jednym z najważniejszych problemów związanych z tworzeniem aplikacji mobilnych jest testowanie. Okazuje się, że testowanie manualne nie jest wystarczająco efektywne i czasami nie gwarantuje zadowalających rezultatów. Kluczowe aspekty, które muszą być uwzględnione w takiej procedurze to: różne rozdzielczości ekranów, różne systemy operacyjne, wersje OS oraz konfiguracja sprzętu. Aby rozwiązać te problemy proponujemy koncepcyjne podejście do półautomatycznego testowania aplikacji mobilnych. W pracy dokładnie opisujemy, jakie metody przetwarzania obrazu mogą być wykorzystane w takim rozwiązaniu. Rozważamy metody segmentacji i dopasowywania szablonów i ich użycie w reprodukcji testów na różnych urządzeniach. Wyniki przeprowadzonych eksperymentów oraz wnioski są również przedstawione i przedyskutowane.

**Słowa kluczowe:** *przetwarzanie obrazów, aplikacje mobilne, framework testowy, segmentacja, dopasowywanie szablonów, k-Means*



# Rozdział VI

## Dlaczego nazwy liczb odczytujemy odwrotnie? Problem reprezentacji liczb

Kazimierz Trzęsicki  
Uniwersytet w Białymstoku

**Streszczenie:** W kulturze europejskiej tylko coś mogło mieć nazwę. Idea nazwania nicości rozwinęła się w Indiach. System arabski, jak nazywamy system dziesiętny, powstał w Indiach. Cyfra „0” ma szczególne miejsce w naszym myśleniu o liczbach. Dziesiętna reprezentacja liczb różni się od standardowego sposobu pisania i czytania w językach europejskich. To dlatego, że system dziesiętny został wprowadzony do Europy przez muzułmańskich uczonych. W językach semickich, takich jak arabski, litery są pisane i czytane od prawej do lewej, a w językach europejskich odwrotnie, czyli od lewej do prawej.

**Słowa kluczowe:** zero, dziesiętna reprezentacja liczb, europejskie a semickie sposoby pisania i czytania

### 1. Wstęp

Sposób reprezentacji wiedzy nie jest obojętny kognitywnie. Jest wiele uzasadnień i argumentów na rzecz tej tezy. Takim argumentem, samym w sobie historycznie ciekawym, jest kwestia reprezentacji liczb.

Kiedy odczytujemy jakiś wyraz, to czytamy go od początku. Gdzie jest początek? To zależy, jak piszemy. Piszemy z lewa na prawo, więc wyraz zaczyna się, ma początek ze swojej lewej strony. Jeśli zasłonimy końcowe litery wyrazu, to zwykle nie ma przeszkód, aby zacząć go odczytywać. Inaczej rzecz ma się z nazwami liczb. Kiedy z jakichś powodów część końcowa nazwy liczby jest zasłonięta, to nie wiemy, jak zacząć ją odczytywać. Na przykład, gdy w nazwie „2021” zasłonięte będą ostatnie dwie cyfry, to jak odczytać pierwsze dwie widoczne? Poniżej damy odpowiedź na to pytanie, umieszczając ją w kontekście historycznym.

## 2. Początki liczenia

Kiedy małe dziecko ma podać liczbę jakichś obiektów, to zwykle pokazywać tyle paluszków, ile jest tych liczonych obiektów. Tak zresztą uczą opiekunowie. Jest to naturalny sposób liczenia.

Pisze Marciszewski [19], że liczenie na palcach było genialnym pomysłem naszych prapradziadów i zauważa, że szympansy, choć do nas podobne, na to nie wpadły, a to daje do myślenia, jeśli chodzi o ludzki umysł. Liczenie na palcach było powszechne w średniowieczu. Tej sztuki nauczano. Instrukcję znajdujemy w ówczesnych podręcznikach, np. Luca Pacioli [27]<sup>12</sup> z 1494 r. Ze stwierdzeniem Marciszewskiego współgra to, co pisze Boecjusz (1867, *De numero lineari*)<sup>13</sup>: Chociaż jedność sama nie jest liczbą liniową, ale ponieważ brakuje jej w ogóle szerokości, jest podstawą poszerzania liczby na długość, również jest początkiem rozciągania liczby na wymiar szerokości. Liczby płaszczyznowe także, chociaż nie są bryłą, nadal są początkiem po uzupełnieniu o wymiar bryły. Liniową liczbą jest ta, która zaczyna się od dwóch z dodawaniem jedności. Do jednej, tej samej i ciągłej linii, dodaje się skumulowane ilości:

*Sic etiam in numero unitas quidem cum ipsa linearis numerus non sit, in longitudine, tamen distenti numeri est principium, et linearis numeris, cum ipse totius latitudinis espres sit, in aliud tamen spatium latitudinis, extenti numeri sortitur initium. Superficies quoque numerorum cum ipsa solidum corpus non sit, addita tamen latitudini altitudine, solidi corporis caput est. Hoc autem planius his exemplis liquebit. Linearis numerus est a duobus inchoans, adiecta semper unitate in unum eundemque ductum quantitas explicata congeries, ut est id quod subiecimus.*

*II. III. IIII. IIIII. IIIIII. IIIIIII. IIIIIIII.*

Wśród podanych nazw liczb nie ma nazwy liczby jeden. Zrozumiemy to, jeśli weźmiemy pod uwagę, że choćby Arystoteles, definiując liczbę jako wielość, nie uznawał jedynek za liczbę. Flamandzki matematyk i inżynier wojskowy Simon Stevin [29], autor *De Thiende* (Sztuka dziesiątek) miał krytyczny stosunek do greckich matematyków – cenił tylko Archimedesa – w szczególności krytycznie odnosił się do tego, że jedność nie jest liczbą, lecz raczej zasadą, *arche*.

Ten związek cyfr z palcami znajduje odzwierciedlenie w języku łacińskim i w tych językach, które poszły jego śladem. Po łacinie „palec” to „digitus”, a angielskie „digit” oznacza palec lub cyfrę.

W języku polskim nazwa „cyfra” oznacza jeden ze znaków: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Nazwa „cyfra” ma dokładnie dziesięć desygnatów. „Cyfra” oznacza więc nie tylko „0”.

<sup>12</sup> <https://www.maa.org/press/periodicals/convergence/mathematical-treasures-paciolis-summa> [20.11.2020].

<sup>13</sup> [https://transcription.si.edu/view/7275/SIL-39088003533734\\_0002](https://transcription.si.edu/view/7275/SIL-39088003533734_0002) [30.11.2020].

W języku angielskim „cipher” oznacza cyfrę hinduską/arabską lub liczbę zero. „Cipher” może być też użyte jako nazwa obliczania, rachowania, szyfrowania, tajnego kodowania, a w odniesieniu do osoby kogoś bez wpływu – kogoś, kto jest zerem.

Zdaniem Mazura [21] nasz mózg jest zaprogramowany na liczenie i podobnie jak ruch palcami jest ono sterowane lewym płatem ciemieniowym. Można przyjąć, że w rozwoju ewolucyjnym pierwotną funkcją lewego płata ciemieniowego było sterowanie palcami, a dopiero później, kiedy kolejne pokolenia liczyły za pomocą palców, ta część mózgu zaprogramowała się na liczenie. Zadajmy pytanie, czy słusznie współczesna szkoła zrezygnowała z nauczania liczenia za pomocą palców. Czy takie palcowe ćwiczenia nie miałyby pozytywnego wpływu na rozwój myślenia matematycznego uczniów? Może jeszcze rodzice uczą dzieci liczyć, wykorzystując patyczki. To jednak nie to samo, co palce. Patyczki niekoniecznie tak aktywują mózg, jak liczenie na palcach.

### 3. Idea zera

Uniwersalny sposób zapisu wymaga znaku na nicłość, pustkę. Zero jest niezbędne dla pozycyjnego zapisu nazw liczb. Zero okazuje się najważniejszą cyfrą. Wynalezienie go było, jak głosi to choćby Bartel Leendert van der Waerden [35]:

*a stroke of genius, to make something out of nothing by giving it a name and inventing a symbol for it.*

*przejawem geniuszu, aby uczynić coś z niczego, nadać mu nazwę i wymyślić jego symbol.*

Zero wymyślili Hindusi. Czy tylko oni? Twierdzi się, że najwcześniejsze, dokładne i systematyczne zastosowanie symbolu zera i zasady pozycjonowania, dokonane zostało przez Majów z Ameryki Środkowej już na początku naszej ery [6].

Jak to się stało, że idea zera zrodziła się w Indiach? Jakies jej ślady znajduje się w pismach babilońskich i greckich. Przedmiotem rozważań pitagorejczyków były kształty i proporcje. Nicość nie ma sensu geometrycznego, nie reprezentuje żadnej figury. Nie pozostaje też w żadnej proporcji do czegokolwiek, a przecież to, co dla Pitagorasa i jego uczniów w kosmosie sensowne, to wzajemne powiązanie przez proporcje. Arystoteles odrzucał istnienie pustki, próżni. Przypisywana jest mu zasada: *horror vacui*, czyli: przyroda nienawidzi próżni.

W *Biblii* istnienie i posiadanie imienia są utożsamiane. Imię może mieć tylko coś, co istnieje. Powołanie do życia dokonuje się głosem. Mojżesz pyta Boga, objawiającego mu się w krzaku gorejącym, o imię. Współplemieńcom mówi, że posłał go Jahwe: Ten, który jest. Nikt nie pyta, czy istnieje. Czy w takiej kulturze mogła się zrodzić idea zera?



W rzymskim systemie zero jest nie tylko niepotrzebne, lecz również kontr-intuicyjne. Nie liczymy, kiedy nic nie ma. Zapis rzymski bazuje na dodawaniu (jedności). Dodawanie zera nic nie zmienia. Liczby są wytwarzane przez jedność, poza zerem: *omnis numerus ab una generatur, ipsa a Nullo*.

Peter Gobets uważa, że:

*Mathematical zero ('shunya' in Sanskrit) may have arisen from the contemporaneous philosophy of emptiness or Shunyata [a Buddhist doctrine of emptying one's mind from impressions and thoughts].*

*Matematyczne zero ('shunya' po sanskrycku) mogło mieć źródło w ówczesnej filozofii pustki lub Shunyata [buddyjska doktryna opróżniania umysłu z wrażeń i myśli].*

Hindusi w swoich koncepcjach religijnych przyznali nicości pozytywny charakter. Mieli ideę nirwany. Te kultury, które nazywały tylko to, co istnieje, nie wpadły na pomysł nazwania czegoś, czego nie ma, co jest pustką. To sanskryckie *sunya* (nicość, pustka) jako nazwa zera daje asumpt do odpowiedniej nazwy w języku arabskim, a następnie łacinie [32].

Dzięki symbolowi zera możliwy stał się pozycyjny liniowy zapis nazw liczb, a to dawało podstawę do algorytmów operacji arytmetycznych. Jak pisze Whitehead [36]:

*One very important property for symbolism to possess is that it should be concise, so as to be visible at one glance of the eye and to be rapidly written. Now we cannot place symbols more concisely together than by placing them in immediate juxtaposition. In a good symbolism therefore, the juxtaposition of important symbols should have an important meaning. This is one of the merits of the Arabic notation for numbers; by means of ten symbols, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, and by simple juxtaposition it symbolizes any number whatever.*

*Bardzo ważną własnością, którą powinna mieć symbolika jest, aby była zwięzła tak, by była postrzegana jednym rzutem oka i była szybko zapisywalna. Ale nie możemy ułożyć razem symboli bardziej zwięzłe niż ustawiając je bezpośrednio jeden po drugim. Dlatego w dobrej symbolice następowanie znaczących symboli powinno mieć znaczący sens. Jest to jedna z wartości arabskiej notacji dla liczb; za pomocą symboli, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 i przez prostą kolejność symbolizujemy dowolną liczbę.*

Zero jest niezbędne dla powzięcia idei liczb ujemnych. Liczby ujemne miały wymiar praktyczny w gospodarce, umożliwiały symboliczną reprezentację długu i pożyczki. Rewolucyjny w rozwoju matematyki jest wynaleziony przez Kartezjusza układ współrzędnych (kartezjańskich). Idea liczby zero jest warunkiem koniecznym tego pomysłu. Wiara w kreatywność pojęcia zera przyświecała założycielom

ZerOrgindia Foundation, <https://www.zerorigindia.org/>. Fundacja działa w przekonaniu, że lepsze rozumienia powstania ZERA prawdopodobnie spowoduje w przyszłości niewyobrażalne innowacje:

*A better understanding of the origin of ZERO, will likely cause more unimaginable innovations in the future.*

*Lepsze rozumienie początków ZERO prawdopodobnie może spowodować niewyobrażalne innowacje w przyszłości.*

## 4. Oswajanie z zerem

Nieintuicyjność zera znajduje wyraz w pojęciu liczby naturalnej. Jest zero liczbą naturalną, czy nie jest? Jedni, głównie logicy, informatycy i matematycy zajmujący się teorią zbiorów, uznają zero za liczbę naturalną, inni nie. We wszystkich językach liczba zero występuje poza systemem językowym: „mam jedno jabłko” brzmi naturalnie, a „mam zero jabłek” jest sztuczne. O wielkim matematyku Wacławie Sierpińskim opowiada się anegdotkę. Sierpiński nie mógł doliczyć się walizek przygotowanych do wyjazdu. „Zginęła jedna walizka” – oświadczył żonie. „Pamiętam, że przyniosłem jedną, a ty powiedziałaś: no dobrze, to już szósta, ostatnia!”. Żona rozejrzała się i stwierdziła: „Są wszystkie”. „Ależ skąd, liczyłem kilka razy: zero, jeden, dwa, trzy, cztery, pięć – a gdzie szósta?”

W chrześcijańskiej Europie – pomijając, że jest to okres wojen krzyżowych, wojen skierowanych przeciw arabskim okupantom Ziemi Świętej – przyjęcie cyfry zero związane było z przełamaniem bariery nazwania czegoś, co oznacza nicność, bowiem „The Ciphra 0 [...] of himselfe signifieth not” (cyfra 0 [...] sama przez się niczego nie oznacza) [9]. A nadto, przecież byłoby to przypadkiem *contradictio in adiecto*. A poza tym Jordan [18]:

*Mußte dies den Romanen und Germanen als etwas Abgeschmacktes erscheinen, so war die orientalische Manier, an diesen scheinbaren Widerspruch mystisch-philosophische Betrachtungen zu knüpfen, bei den Abendländern geeignet, geradezu Verdacht zu erwecken: ein Zauber sei im Spiele. Der Kodex des Klosters Salem schreibt nämlich: „Jede Zahl entsteht aus der Eins, jene aber aus der Null.”*

*musiało się to Romanom i Germanom zdawać nieco absurdalne, ponieważ właściwa ludzom Zachodu orientalna maniera wiązania mistyczno-filozoficznych rozważań z tą pozorną sprzecznością tym samym budziła podejrzenie: magia jest w tej grze. W Kodeksie klasztoru z Salem jest mianowicie napisane: „Každa liczba powstaje z jedyńki, ta zaś z zera.”*

W ówczesnej kulturze rozumienia liczb mocno zdominowanej – co wciąż dziedczymy – przez tradycję pitagorejską i kabałę, uznaniu zera za liczbę towarzyszyły spekulacje na temat jego pozamatematycznego znaczenia. Pitagoras był czczony jako mędrzec, mąż stanu i cudotwórca. To od „μαθηματικοί” (*mathematikoi*), co po grecku dosłownie znaczy „uczeni”, nazwy najbardziej wtajemniczonej grupy założonego przez niego bractwa religijno-mistycznego, pochodzi dzisiejsza nazwa matematyki.

Świadectwem oddziaływania myśli żydowskiej są średniowieczne katedry [28]:

*It is perhaps one of the stranger ironies of history that this originally Jewish yearning for transcendence and numinous mathematics would find its greatest architectural expression in the towering Gothic cathedrals built by Christian stonemasons.*

*Jest to być może jedna z nieznanych ironii historii, że pierwotna żydowska tęsknota za ponadziemską i świętą matematyką znalazła swój największy architektoniczny wyraz w strzelistych gotyckich katedrach budowanych przez chrześcijańskich kamieniarzy.*

Więcej, uważa się, że budowniczowie gotyckich katedr [1]:

*employed an eastern treasure of sacred geometry drawn from the Pythagorean school,*

*zastosowali wschodnie bogactwo świętej geometrii wywodzącej się ze szkoły Pitagorasa,*

którą pozyskali z żydowskiej tradycji. Arytmetyczne proporcje znalazły odzwierciedlenie w architekturze katedr [20].

Szczególna pozycja zera i jedynki daje racje dla różnych wyjaśnień. Czytamy [26]):

*One the striking aspects of Stevin's Arithmetique (1585) is the way the zero takes over many of the traits formerly assigned to the one. In their respective concepts, the one and the zero are the origin of number, but no number; they are also declared to be the arithmetical counterpart of the geometrical point.*

*Jednym z uderzających aspektów Arithmetique (1585) Stevina jest to, że zero przejmuje wiele cech charakterystycznych uprzednio przypisywanych jedynce. W ich odpowiednich pojęciach, jedynka i zero są początkiem liczby, lecz nie liczbą; są one także ogłaszane jako odpowiednik punktu geometrycznego.*

Dla neoplatoników Bóg, Jedność, jest początkiem rzeczy, ale nie jest rzeczą, jak jedynka nie jest liczbą, ale początkiem liczb.

„Podejrzliwość” wobec zera znajdowała wyraz w kulturze. Nie brakowało różnego rodzaju tekstów ośmieszających [18]. Przykładem może być fragment wiersza Lubeca, którego autorem jest Henricus Aquipolensis:

*Ut pupa praesumpsit aquila esse, asinus leo quondam,  
Simia regnatix – Cifra figura fore.*

*Jak marionetka chciałaby być orłem, osioł lwem,  
Małpa królową – tak pustka chciałaby być znakiem.*

Zapoznaniu się z cyframi hinduskimi (arabskimi) – mimo, że jest to system dziesiętny – towarzyszy fascynacja zerem. Zeru nadaje się jakiś wymiar metafizyczny i teologiczny. W napisanym ok. 1143 r. [23] Kodeksie z Salem [7] czytamy:

*Nec praetereundum est quod 0 per omnia omnibus algorizmi utitur legibus quemadmodum et alia figura, excepto quod nullum numerorum multiplicat, sed et ipsa a nullo multiplicatur. Quid enim [aliud] si dixeris milies nichil quam nichil? Aut nichil ad mille quam mille? Facit tamen quandam multiplicationem, sed tandem per decuplationem: Verba gratia praepone 0 uni, et fiunt X, praepone deceno et fiunt C., praepone centeno et erunt M. Et sciendum, quod in hoc magnum latet sacramentum. Per hoc, quod sine inicio est et fine, figuratur ipse, qui est vere alpha et  $\omega$ , id est sine inicio et fine; et sicut 0 non auget nec minuit, sic ipse nec recipit augmentum nec detrimentum; et sicut omnes numerus decuplat, sic ipse non solum decuplat, sed millificat, immo ut verius dicam omnia ex nichilo creat, conservat atque gubernat.*

Nie powinniśmy pominąć faktu, że 0 jest przedmiotem reguł algorytmicznych jak są inne cyfry, oprócz tego, że nie prowadzi do powiększenia żadnej liczby, a i samo nie powiększa się przez żadną liczbę. Jak faktycznie nazwalibyśmy tysiąckrotność nicości, jak nie nicością? Zaś nic dodane do tysiąca, jak nie tysiącem? Zero, jednakże, dokonuje pewnego rodzaju wzrost, lecz tylko przez zwielokrotnienie przez dziesięć: zatem, na mocy Słowa, połóż zero przed jeden, a otrzymasz dziesięć, połóż przed dziesięć, a otrzymasz sto, połóż przed sto, a otrzymasz tysiąc. I trzeba wiedzieć, że wielka w tym kryje się tajemnica. Przez to, co jest bez początku i końca on sam siebie ukazuje jako ten, który prawdziwie jest alfa i  $\omega$ , to jest bez początku ani końca, i tak, jak 0 nie rośnie ani nie maleje, tak on sam ani nie powiększa się, ani nie doznaje uszczerbku; i, jak ono wszystkie liczby udziesięciokrotnia, tak on sam nie tylko udziesięciokrotnia, ale utysiąckrotnia, co więcej, abym bardziej prawdziwie się wyraził, wszystko tworzy z niczego, zachowuje i kieruje.

## 5. Cyfra zero

Powyższy cytat inspiruje pytanie o kształt cyfry zero. Nie zaskakuje, że cyfra ta ma kształt okręgu – przecież nic lepiej nie obrazuje czegoś bez początku i bez końca, może kropka. Okrąg jednak lepiej wyraża pustkę. Według hinduskiego mistycyzmu zero jest okrągłe, bo oznacza okrąg życia, lub co było znane jako „wąż wieczności”.

Podając kwestię symbolu zera nie unikniemy pytania o kształt litery „o”. Zresztą zdarza się nam pomylić jedno z drugim, literę z cyfrą. Są różne powody, dla czego jedno i drugie mają podobne symbole. Amerykanie w niektórych kontekstach odczytują cyfrę zero jak literę „o”. Nasze cyfry arabskie mają podobne symbole jak w sanskrycie. Okrąg już w sanskrycie był symbolem nicości, zera, choć początkowo zero oznaczano za pomocą kropki. Nie znaczy to, że nie było żadnych innych symboli. Kształty cyfr, w tym zera, były zmieniane [6].

Na dwa wieki przed naszą erą Grecy astronomowie stosowali sześćdziesiątkowy system babiloński, co dziedziczymy do dzisiaj, posługując się tym systemem, w szczególności w miarach czasu. Zapisując liczby używali litery omikron do oznaczania pustej przestrzeni. W Bizancjum było to zwykle  $\omicron$  [6].

Greckie „o” wywodzi się z fenickiej litery *ayin*, a ta z arabskiej. Jej kształt był inspirowany egipskim hieroglifem oka. Grecy adaptowali ten znak na greckie małe „o” – omikron. Znakiem wielkiego „O” – omegi jest:  $\Omega$ . „ $\Omega$ ” wywodzi się z podwójnego omikron:  $\omega$ . Omikron było też znakiem liczby 70. Kształt „ $\Omega$ ” miałby wzorzec w układzie ust, kiedy głoska ta jest wymawiana. „ $\Omega$ ” była też znakiem liczby 800.

Badania okręgu inspirowały rozwój geometrii, rachunku, astrologii, a w różnych okresach traktowano okrąg jako sprawę boską, wiązano z wiedzą tajemną chronioną nawet karą śmierci [17]. Kojarzenie boskości z okręgiem było obecne w kulturze wielu ludów.

Przytoczony fragment *Kodeksu z Salem* może wskazywać na to, że brak akceptacji na nazwanie czegoś, czego materialnie nie ma, zniwelowano, wiążąc z zerem coś duchowego. Zero jest „kreatywne”, bo ma wymiar duchowy.

John von Neumann [34], mając zbiór pusty i operację tworzenia zbiorów ze zbiorów, zdefiniował liczby naturalne.

Współczesna fizyka głosi, że energia grawitacyjna ma przeciwny znak niż wszystkie inne rodzaje energii. Jeśli więc energia grawitacyjna całości kosmosu dokładnie równoważy energię związaną z masami ciał kosmicznych i innymi rodzajami energii, to łączna wartość energii wszechświata jest zerowa. Przed stworzeniem i po stworzeniu bilans energii jest zerowy. Wszechświat może kreować się z nicości. Pierwszym rozważającym tę możliwość był Edward Tryon z City University w Nowym Yorku, który pytał: „Is the Universe a Vacuum Fluctuation?” [31]. Stephen Hawking i Leonard Młodinow w *The Grand Design* objaśniają powstanie świata bez hipotezy Boga. Piszą [15]:

*Because there is a law such as gravity, the universe can and will create itself from nothing. Spontaneous creation is the reason there is something rather than*

*nothing, why the universe exists, why we exist. It is not necessary to invoke God to light the blue touch paper and set the universe going.*

*Ponieważ jest prawo takie, jak grawitacja, wszechświat może i będzie tworzyć sam siebie z niczego. Spontaniczna kreacja jest racją tego, że jest raczej coś niż nic, dlatego istnieje wszechświat, dlatego my istniejemy. Nie jest konieczne przywoływanie Boga, aby zapalić lont i poruszyć wszechświat.*

W starożytnej Grecji liczba – ἀριθμός (arithmós) była pojmowana jako skończona wielość złożona z jednostek [3, 11].

Greckie myślenie matematyczne było geometryczne. Jako liczby nie pojmowali nie tylko nicości, lecz również jedności. Liczba była wielością. Kiedy mówimy po polsku, że jest czegoś wiele, to rozumiemy, że tego czegoś są przynajmniej dwa egzemplarze.

Można przypuszczać, choćby w oparciu o jeszcze współcześnie trwające kultury plemienne, w których językach są nazwy na liczby jeden i dwa, a nie ma nazw liczb większych, innych niż „wiele”, że w ogóle człowiek przez długie początkowe lata swojego rozwoju miał tylko te trzy określenia na ilość: jeden, dwa, wiele. Liczba trzy była odkryciem tak znaczącym, że zyskała szczególne miejsce w kulturze.

W liczbie trzy od starożytności dopatrywano się jakiejś doskonałości. Posługiwano się nią przy ofiarach i modłach. Przytoczmy fragment wywodów Arystotelesa zawartych w *O niebie* [2]:

*„Ciałem” (σῶμα) jest to, co jest podzielne we wszystkich wymiarach. Ta spośród wielkości, która rozciąga się w jednym wymiarze, jest linią; ta, która rozciąga się w dwóch wymiarach, jest powierzchnią; ta, która rozciąga się w trzech wymiarach, jest ciałem. Prócz tych nie ma żadnej innej wielkości, bo liczba „trzy” obejmuje wszystko, a „trzy razy” znaczy tyle co „całkowicie”. W rzeczy samej, jak mówią pitagorejczycy, cały świat i wszystkie rzeczy w nim zawarte są określone liczbą „trzy”; koniec, środek i początek tworzą liczbę, która cechuje „całość”, a liczba ta jest „triadą”. Ponieważ uzyskaliśmy tę liczbę od natury, jak gdyby stanowiła ona jedno z jej praw, dlatego posługujemy się nią także w kulcie bogów.*

*W podobny sposób posługujemy się w określaniu rzeczy, bo gdy mowa o dwóch przedmiotach, mówimy „oba”; gdy chodzi o dwóch ludzi, mówimy „obaj”; a nie „wszystkie”, „wszyscy”. Tym ostatnim wyrazem zaczynamy się posługiwać dopiero wtedy, gdy mamy do czynienia z przynajmniej trzema rzeczami.*

Zauważmy, że we współczesnym języku potocznym określenie czegoś lub kogoś jako zera jest pejoratywne.

## 6. System dziesiętny

Na dwa wieki przed naszą erą greccy astronomowie stosowali sześćdziesiątkowy system babiloński [6]. Joseph Mazur [21] pyta, dlaczego Grecy mimo swej błyskotliwości nie dostosowali systemu babilońskiego z jego pozycyjnością i relatywną łatwością zapisu liczb. Babilończycy mieli jasną ideę notacji pozycyjnej i użycia tych samych cyfr dla reprezentacji różnych potęg 60. I odpowiada, że być może dlatego, że do liczenia używali abaka lub być może dlatego, że byli bardziej zainteresowani okazałością zakresu samej matematyki niż samymi obliczeniami. Zauważa też, że [21]:

*The difficulty is in distinguishing placeholder from number. Accepting zero as a number representing the absence of quantity would have been a fantastically daring idea.*

*Trudność znajduje się w odróżnieniu pozycji miejsca od liczby. Akceptacja zera jako liczby reprezentującej brak ilości była fantastycznie śmiałą ideą.*

Zdaniem Johna Barrowa [4] hinduski system rachowania jest prawdopodobnie najbardziej owocną intelektualną innowacją ludzkości w ogóle. Richard Fuller [13] twierdzi, że:

*The fifteenth-century Europeans' adoption of Arabic numerals and their computation-facilitating "positioning-of-numbers" altogether made possible Columbus's navigational calculations and Copernicus's discovery of the operational patterning of the solar system and its planets. Facile calculation so improved the building of the ships and their navigation that the ever-larger ships of the Mediterranean ventured out into the North and South Atlantic to round Africa and reach the Orient. With Magellan's crew's completion of his planned circumnavigation, the planet Earth's predominantly water-covered sphericity was proven. The struggle for supreme mastery of human affairs thus passed out of the Mediterranean and into the world's oceans.*

Przejęcie w XV w. przez Europejczyków arabskiej numeracji i jej rachunkowych ułatwień przez „pozycjonowanie liczb”, ostatecznie umożliwiło Kolumbowi obliczenia nawigacyjne oraz Kopernikowi odkrycie uwarunkowań działania systemu słonecznego i jego planet. Ułatwione obliczenia polepszyły budowę statków i ich nawigację tak, że coraz większe statki ze Śródziemnomorza wyprawiały się na Północny i Południowy Atlantyk dookoła Afryki i osiągały Orient. Wraz z zakończeniem przez załogę Magellana zaplanowanego opłynięcia, została dowiedziona kulistość planety Ziemia, która przeważnie pokryta jest wodą. Walka o najwyższe panowanie ludzkich spraw przeszła przez Śródziemnomorze na światowe oceany.

Początki stosowania w Europie systemu dziesiętnego wiązane są z Gerbertem z Aurillac (ok. 946–1003), późniejszym papieżem Sylwestrem II (był nim od 999 r.

do swojej śmierci) [12]. Sylwester II m.in. ustanawiał organizację kościelną na ziemiach Polan. Gilbert kształcił się w klasztorach w Katalonii, gdzie były dostępne rękopisy muzułmańskie, w szczególności z Kordoby, ówczesnego centrum intelektualnego. Jego nauczycielem miał być Żyd Josephus Hispanus [18]. Z powodu umiejętności obliczania był posądzany o konszachty z diabłem, co znalazło wyraz w rycinie z 1460 r. W *Materialien zur Geschichte der arabischen Zahlzeichen in Frankreich* [18] czytamy:

*Ja Wilhelm [von Malmesbury] erklärt selber: „Aber dies alles möchte man für Erfindungen halten, weil doch das Volk gewohnt ist, den Ruhm der Gelehrten zu beschmutzen, indem sie von jedem, der etwas außerordentliches leistet, behaupten, er habe mit einem Dämon Verkehr gehabt.“ Niemand anders soll nämlich dem späteren Papste die Kunst, mit arabischen Zeichen zu rechnen, gezeigt haben, als der Teufel. Und man nennt sie ja auch heute noch die „Teufelskunst“.*

*Wszak Wilhelm [z Malmesbury] sam wyjaśnia: „Zaś to wszystko można by uznać za wynalazek, jednak przecież lud zwykł sławę uczonych zmącić, stąd każdego, kto uzyskał coś nadzwyczajnego, uznaje, że ma konszachty z demonem.“ Nikt inny nie powinien późniejszemu papieżowi pokazać sztuki rachowania arabskimi znakami jak tylko diabeł.*

Do rozpowszechnienia arabskiego zapisu przyczynili się ludzie interesu. Pisze Jordan [18]:

*Dreimal hat der Kaufmannsstand in der Geschichte der arabischen Zahlzeichen in romanischen Ländern eine entscheidende Rolle gespielt. Es ist wahrscheinlich, daß er beim ersten Import aus Spanien die Hand mit im Spiele gehabt hat. – Leonardo Pisano war Kaufmann, der praktische Zweck hat in seinem Lehrbuch an erster Stelle gestanden und hat ihm für die Dauer des Mittelalters in Italien die Herrschaft gesichert. – In Frankreich waren es am Ausgang des Mittelalters wiederum Kaufleute, die eine entstandene Lücke ausfüllten, gleichsam die arabischen Zahlzeichen für ihren Eroberungsgang im 16. Jahrhundert noch fertig machten.*

*Da verschwand dann alles, was ihnen an Komischem und Mystischem angehangen hatte, nur wenige Ausdrücke und Redensarten blieben bestehen, als fossile Reste aus einer vergangenen, wenig kritisch angelegten Zeit.*

*W historii arabskich cyfr w krajach romańskich trzykrotnie kupiectwo odegrało decydującą rolę. Prawdopodobne jest, że miało swój udział w pierwszym ich pozyskaniu z Hiszpanii. – Leonardo Pisano był kupcem, jego podręcznik na pierwszym miejscu stawiał cel praktyczny, co zapewniło mu trwałą powszechność w średniowiecznej Itali. – We Francji u schyłku średniowiecza znowu kupcy,*



*którzy wypełnili powstałą lukę, jakby uczynili gotowymi cyfry arabskie na swoje przejście w 16. wiek.*

*Tak przepadło wtedy wszystko, co doczepiono im komicznego i mistycznego, tylko pozostały nieliczne wyrażenia i zwroty, jak skamieniałe pozostałości mało krytycznie ukierunkowanego czasu, który przeminął.*

Udział ludzi praktyki miał istotne znaczenie nie tylko dla rozpowszechnienia cyfr arabskich, ale i wytworzenia nowego paradygmatu matematyki skoncentrowanego na operacjach na liczbach, a nie na samych liczbach.

Ludzie nauki, z powodu ówczesnego paradygmatu matematyki, spekulatywnej arytmetyki nauczanej w ramach *quadrivium* – zainteresowani przedmiotami raczej niż operacjami matematycznymi, a także powiązaniami kabalistycznymi – pozostawali przy systemie rzymskim, a ten nie inspirował zainteresowania operacjami. Nowy paradygmat matematyki formowały prace inżynierów, takie jak np. *L'Arithmetique* [30] Simona Stevina oraz *An Arithmetical Warlike Treatise Named Stratoticos* [9] Leonarda i Thomasa Diggesów. Teksty arytmetyki praktycznej pisane były w języku rodzimym, co nie znaczy, że praktycy nie czynili tego bez oporów. Przyzwyczajenia są oczywistą przeszkodą, ale przejście na system dziesiętny wymagało też zmiany jednostek pieniężnych oraz jednostek wag i miar. Na przykład system monetarny Wielkiej Brytanii został zmieniony na dziesiętny dopiero w 1970 r.

## 7. Zapis liczb

System pozycyjny Europa w średniowieczu przejęła od Hindusów za pośrednictwem Arabów. Fibonacci (Leonardo Bonacci, Leonardo z Pizy lub Leonardo Bigollo Pisano) o nauce pobieranej na terenie obecnej Algierii pisał [24]:

*Ubi ex mirabili magisterio in arte per novem figuris Indorum introductus.*

*Gdzie przez wspaniałego nauczyciela zostałem wprowadzony w sztukę za pomocą dziewięciu cyfr hinduskich.*

W *Liber Abaci* (1202, Cap. I) czytamy:

*Novem figure Indorum he sunt 9 8 7 6 5 4 3 2 1. Cum his itaque novem figuris, et cum hoc signo 0, quod arabic cephirum appellatur, scribitur quilibet numerus.*

*Dziewięć cyfr hinduskich to 9 8 7 6 5 4 3 2 1. Z nimi i ze znakiem 0, który Arabowie nazywają ciphurum, dowolna liczba może być napisana.*

W *Carmen de Algorismo lub Algorismus metricus*, wierszu łacińskim dla potrzeb dydaktyki przypisywanym Aleksandrowi de Villa Dei (Alexander de Villedieu), czytamy:

*Hinc incipit algorismus.  
Haec algorismus ars praesens dicitur in qua  
Talibus Indorum fruimur bis quinque figuris  
0 9 8 7 6 5 4 3 2 1,*

*Tu zaczyna się algorytm.  
Ta nowa sztuka jest nazwana algorytmem, w którym  
z tych dwukrotnie pięciu cyfr  
0 9 8 7 6 5 4 3 2 1,*

od Hindusów czerpiemy taką korzyść.

W obu cytatach kolejność wskazywania cyfr jest różna do tej, do której jesteśmy przyzwyczajeni. Jest tak dlatego, że Arabowie, jak to jest u Semitów, pisali (i czytali) z prawa na lewo. Wraz z cyframi hinduskimi od Arabów przejęliśmy również semicki sposób ich reprezentacji, a zapisujemy tak, jak jest to dla nas zwyczajne: z lewa na prawo. A co z odczytywaniem? W języku polskim odczytujemy po „semicku” liczby dwucyfrowe do dwudziestu i także końcówki innych liczb, a więc: 11 – jedenaście, czyli jeden dziesięć; 12 – dwanaście, czyli dwa dziesięć, itd. Powyżej dwudziestu jest już: 21 – dwadzieścia jeden, 22 – dwadzieścia dwa, itd.

W języku niemieckim taka praktyka obejmuje jeszcze liczby od 21 do 99 oraz ich odczytywanie jako końcówek nazw większych liczb. W Turcji po reformach Atatürka, nazwy liczb, również tych od 11 do 19, odczytywane są zgodnie z kolejnością cyfr z lewa na prawo [33], czyli: 11 – dziesięć jeden; 12 – dziesięć dwa, itd.

Dlaczego w języku niemieckim nazwy liczb dwucyfrowych i ostatnie dwie cyfry nazw liczb większych są odczytywane w innej kolejności niż są pisane, czyli – inaczej mówiąc – dlaczego semicki sposób odczytywania ma szerszy zakres [14]?

W pierwszym drukowanym podręczniku w języku niemieckim wydanym w 1482 r. jego autor nie podał sposobu odczytywania arabskich nazw liczb dwucyfrowych. W książce Jakoba Köbela wydanej w 1517 r. [16] podawany jest sposób odczytywania takich nazw, np. 21 – „zwentzigeins”, „zwanzigeins” [14]. Rozstrzygnięcia dokonał Marcin Luter [14]:

*In der Übersetzung der Bibel ins Deutsche schloss sich Luther dem Vorschlag von Köbel für Zahlenbenennungen nicht an, wodurch sich die verdrehte Zahlensprechweise etabliert hat. Man kann aber annehmen, dass Köbel versucht hat, Luther für seine Zahlensprechweise zu gewinnen. Auch Adam Ries, der ab 1522 zahlreiche Rechenbücher herausbrachte, blieb bei der verdrehten Sprechweise. [...] Aus diesem Gründen blieb es bei einer unvollständigen Reform,*

*in der nur die Schreibweise, aber nicht die Sprechweise von Zahlen verändert wurde.*

*Kiedy Luter przekładał Biblię na niemiecki nie zgodził się z propozycją Köbele w sprawie nazw liczb, dlatego ustalił się odwrotny sposób wymawiania liczb. Można przyjąć, że Köbel próbował namówić Lutra na swój sposób wymawiania liczb. Także Adam Ries, który od 1522 r. wydawał liczne podręczniki rachunków, pozostał przy odwrotnym sposobie mówienia. [...] Z tych powodów nie została skończona reforma, która miała zmienić nie tylko sposób pisania, lecz także mówienia.*

Dodajmy, że Lothar Gerritzen założył *Zwanzigeins*<sup>14</sup>, towarzystwo na rzecz zmiany dotychczasowego sposobu odczytywania nazw liczb w języku niemieckim.

Rozumienie zapisu:

$$a_n a_{n-1} \dots a_0$$

dokonuje się według wzorca:

$$a_n 10^n + a_{n-1} 10^{n-1} + \dots + a_0 10^0,$$

a zatem w wykładnikach (na pozycjach potęg dziesięciu) jest porządek wyznaczony kolejnością liczb z prawa na lewo. Pełna realizacja porządku z lewa na prawo – w jakim zwykliśmy czytać – byłaby, gdyby kolejność cyfr w zapisie była dokonana według wzorca, jaki stosuje się do wypowiadania (ale nie do zapisywania) liczb dwucyfrowych w języku niemieckim:

$$a_0 10^0 + a_1 10^1 + \dots + a_n 10^n.$$

Taki zapis byłby zgodny z „niemieckim” sposobem wypowiadania nazw liczb dwucyfrowych i dwucyfrowych końcówek. „24” wypowiadalibyśmy: dwa i czterdzieści. A „356” wypowiadalibyśmy: trzy i pięćdziesiąt, i sześćset. Zapis ten nie zmuszałby do zabiegu, który musimy wykonać dla obecnego zapisu, a mianowicie policzenia ilości cyfr. W zapisie tym potęga dziesiątki byłaby zgodna z kolejnością w zapisie z lewa na prawo, a nie odwrotnie, jak jest obecnie (i jak zostanie).

## 8. Zakończenie

Reprezentacja liczb w systemie arabskim/hinduskim okazała się istotna dla rozwoju matematyki. Zadajemy pytanie, czy to że liczby zapisujemy po semicku, choć fizycznie

<sup>14</sup> <https://zwanzigeins.jetzt> [24.10.2020].

piszemy z lewa na prawo, a odczytujemy według ogólnej zasady tak, jak zapisujemy, ma jakiś wymiar kognitywny? Czy ma jakiś wpływ na nasze rozumienie i postrzeganie liczb? Zdaje się, że jest to bez znaczenia.

Konsekwencją przejścia semickiego sposobu zapisu jest sposób wykonywania algorytmów (pisemnego) dodawania, mnożenia itd. Argumenty liczbowe podpisujemy pod sobą zgodnie z potęgą dziesięciu od prawej strony na lewą, czyli zgodnie z semickim sposobem pisania. Wydaje się, że ta „odwrotność” algorytmu nie ma praktycznych konsekwencji.

Czy sposób zapisu liczb ma jakieś konsekwencje praktyczne? Jedyne, co przychodzi na myśl, to prezentowanie cen. Jeśli weźmiemy pod uwagę, że bardziej zwracamy uwagę na to, co postrzegamy pierwsze, to możemy uznać, że typowe w handlu zaniżanie pierwszej cyfry w nazwie liczby może sprzyjać (nieuświadomianemu) przekonaniu, że cena oferowanego produktu jest niższa, niż faktycznie jest. Taki efekt uzyskuje się, kiedy np. „2000” zastąpione jest przez „1999” – trudniej podejmuje się decyzję o zakupie produktu za dwa tysiące, a łatwiej za tysiąc „z groszami”.

## Literatura

- [1] Åkerman, S. (1998). *Rose Cross over the Baltic: The Spread of Rosicrucianism in Northern Europe*. Leiden: Brill.
- [2] Arystoteles. (1980). *O niebie*. Warszawa: PWN. (Przełożył, wstępem, komentarzem i skorowidzem opatrzył Paweł Siwek).
- [3] Arystoteles. (1983). *Metafizyka*. Warszawa: PWN. (Przełożył, wstępem, komentarzem i skorowidzem opatrzył Kazimierz Leśniak).
- [4] Barrow, J. D. (2000). *The Book of Nothing: Vacuums, Voids, and the Latest Ideas about the Origins of the Universe*. London: Vintage Books. <https://archive.org/details/indexcriticism00glor>.
- [5] Boetii, A. M. T. S. (1867). *De institutione arithmetica* (Vols. 1–2; G. Friedlein, Ed.). Lipsiae: B. G. Teubneri. [https://la.wikisource.org/wiki/De\\_Arithmetica](https://la.wikisource.org/wiki/De_Arithmetica), [https://www.documenta-catholicaomnia.eu/03d/0480-0524,\\_Boethius,\\_De\\_Arithmetica,\\_LT.pdf](https://www.documenta-catholicaomnia.eu/03d/0480-0524,_Boethius,_De_Arithmetica,_LT.pdf).
- [6] Cajori, F. (1993). *A History of Mathematical Notations*. New York: Dover Publications, Inc. (Two volumes bound as one. Volume I: Notations in Elementary Mathematics. Volume II: Notations Mainly in Higher Mathematics).
- [7] Cantor, M. (1865). *Über einen Codex des Klosters Salem*. In: *W: Zeitschrift für Mathematik und Physik*, 10, 1–16. <https://archiv.ub.uni-heidelberg.de/volltextserver/12869/>.
- [8] Digges, L. (1579). *An Arithmetical Warlike Treatise, Named Stratoticos*. London: Henrie Bynneman.
- [9] Digges, L., & Digges, T. (1590). *An Arithmetical Warlike Treatise Named Stratoticos* (see (Digges, 1579) ed.). London: Richard Field.
- [10] Fibonacci, L. (1202). *Liber abbaci*. [https://la.wikisource.org/wiki/Liber\\_abbaci](https://la.wikisource.org/wiki/Liber_abbaci).
- [11] Fitzpatrick, R. (Ed.). (2008). *Euclid's Elements of Geometry* (First edition – 2007; Revised and corrected – 2008 ed.). <https://farside.ph.utexas.edu/Books/Euclid/Elements.pdf>. (The Greek text of J.L. Heiberg (1883–1885) from *Euclidis Elementa*, edidit et Latine interpretatus est I. L. Heiberg, in aedibus B. G. Teubneri, 1883–1885; edited, and provided with a modern English translation, by Richard Fitzpatrick).

- [12] Friedlein, G. (1861). *Gerbert, die Geometrie des Boethius und die indischen Ziffern: Ein Versuch in der Geschichte der Arithmetik*. Erlangen: Verlag von Theodor Blaesing. <https://ia800205.us.archive.org/28/items/gerberdiegeome01dostgoog/gerberdiegeome01dostgoog.pdf>.
- [13] Fuller, R. B. (1982). *Critical Path*. San Francisco, CA: Estate of R. Buckminster Fuller. (Contributor: Kiyoshi Kuromiya).
- [14] Gerritzen, L. (2008). Warum wir Zahlen von hinten nach vorne lesen und warum das nicht so bleiben muss. In L. Gerritzen (Ed.), *Zwanzigeins: Für die unverdrehte Zahlensprechweise. Fakten, Argumente, Meinungen* (pp. 22–33). Bochum: Brockmeyer Verlag. [https://zwanzigeins.jetzt/downloads/Gerritzen%20et%20al%20\(2008\)%20-%20Zwanzigeins.pdf](https://zwanzigeins.jetzt/downloads/Gerritzen%20et%20al%20(2008)%20-%20Zwanzigeins.pdf).
- [15] Hawking, S., & Mlodinow, L. (2010). *The Grand Design*. New York: Bantam Books.
- [16] Hergenbahn, R. (2008). Die Köbelschen Zahlentafeln in seinen Rechenbüchern. In L. Gerritzen (Ed.), *Zwanzigeins: Für die unverdrehte Zahlensprechweise. Fakten, Argumente, Meinungen* (pp. 109–112). Bochum: Brockmeyer Verlag. [https://zwanzigeins.jetzt/downloads/Gerritzen%20et%20al%20\(2008\)%20-%20Zwanzigeins.pdf](https://zwanzigeins.jetzt/downloads/Gerritzen%20et%20al%20(2008)%20-%20Zwanzigeins.pdf).
- [17] Adams, T. E., Jones, S. H. & Ellis, C. (Eds.). (2016). *Handbook of Autoethnography*. London and New York: Routledge.
- [18] Jordan, L. (1905). Materialien zur Geschichte der arabischen Zahlzeichen in Frankreich. *Archiv für Kulturgeschichte*, 3(2), 155–195. <https://archive.org/details/materialenzurges00jord/mode/2up>.
- [19] Marciszewski, W. & Stacewicz, P. (2011). *Umysł – komputer – Świat. O zagadce umysłu z informatycznego punktu widzenia* (prof. Leonard Bolc, Ed.). Warszawa: Akademicka Oficyna Wydawnicza EXIT.
- [20] Masi, M. (1983). *Boethian Number Theory: A Translation of the De Institutione Arithmetica (with Introduction and Notes)*. Amsterdam – New York: Rodopi.
- [21] Mazur, J. (2014). *Enlightening Symbols: A Short History of Mathematical Notation and Its Hidden Powers*. Princeton: Princeton University Press.
- [22] Menninger, K. (1934). *Zahlwort und Ziffer: Aus der Kulturgeschichte unserer Zahlsprache, unserer Zahlschrift und des Rechenbrettes*. Breslau: Ferdinand Hart.
- [23] Menninger, K. (1958). *Zahlwort und Ziffer: eine Kulturgeschichte der Zahl* (2<sup>nd</sup> ed., Vols. 1–2). Göttingen: Vandenhoeck & Ruprecht. (First edition: (Menninger, 1934)).
- [24] Menninger, K. (1969). *Number Words and Number Symbols: A Cultural History of Numbers*. Cambridge, MA: M.I.T. Press. (Translation of (Menninger, 1958)).
- [25] O'Connor, J. J., & Robertson, E. F. (2012). *Gerbert of Aurillac*. In: *MacTutor History of Mathematics*. School of Mathematics and Statistics University of St Andrews, Scotland. <https://mathshistory.st-andrews.ac.uk/Biographies/Gerbert/>.
- [26] Ostashevsky, E. (2000). *Quintessence from Nothingness: Zero, Platonism, and the Renaissance* (a dissertation for the degree of Doctor of Philosophy, Stanford University, 300 North Zeeb, Ann Arbor, MI). [https://www.academia.edu/43659945/Quintessence\\_from\\_nothingness\\_9986171\\_978\\_0\\_599\\_93154\\_1](https://www.academia.edu/43659945/Quintessence_from_nothingness_9986171_978_0_599_93154_1).
- [27] Pacioli, L. (1523). *Summa de arithmetica geometria proportioni: et proportionalita...* Paganino de Paganini. Retrieved from <https://books.google.pl/books?id=iqgPe49fhrsC>.
- [28] Schuchard, M. K. (2002). *Restoring the Temple of Vision: Cabalistic Freemasonry and Stuart Culture*. Leiden-Boston-Köln: Brill.
- [29] Stenlund, S. (2014). *The Origin of Symbolic Mathematics and the End of the Science of Quantity* (Vol. 59). Uppsala: Uppsala Universitet.

- [30] Stevin, S. (1585). *L'Arithmetique*. Leyde: De l'Imprimerie de Christophle Plantin. [https://books.google.pl/books?id=1dU5AAAACAAJ&printsec=frontcover&source=gbs\\_ge\\_summary\\_r&cad=0#v=onepage&q&f=false](https://books.google.pl/books?id=1dU5AAAACAAJ&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false).
- [31] Tryon, E. P. (1973). Is the Universe a Vacuum Fluctuation? *Nature*, 246, 396–397. doi: 10.1038/246396a0.
- [32] Trzęsicki, K. (1987). Rola pojęcia niebytu w twórczości matematycznej. *Idea. Studia nad strukturą i rozwojem pojęć filozoficznych*, 2, 75–85. (Czarnawska, M. and Kopania, J. (red.)).
- [33] Voigt, J. (2008). Die konsequente Zahlensprechweise in der Türkei. In L. Gerritzen (Ed.), *Zwanzigeins: Für die unverdrehte Zahlensprechweise. Fakten, Argumente, Meinungen* (pp. 113–114). Bochum: Brockmeyer Verlag. [https://zwanzigeins.jetzt/downloads/Gerritzen%20et%20al%20\(2008\)%20-%20Zwanzigeins.pdf](https://zwanzigeins.jetzt/downloads/Gerritzen%20et%20al%20(2008)%20-%20Zwanzigeins.pdf).
- [34] von Neumann, J. (1922–23). Zur Einführung der transfiniten Zahlen. *Acta Litt. Ac. Asc. Hung. Fran. Joseph.*, 1, 199–208.
- [35] Waerden, B. L. Van der. (1961). *Science Awakening*. New York: Oxford University Press. (English translation by Arnold Dresden, with additions of the author.)
- [36] Whitehead, A. N. (1911). *An Introduction to Mathematics*. London: Williams and Norgate. <https://archive.org/details/introductiontoma00whitalia>.

***Why do we read the names of numbers the other way around? The number representation problem***

**Abstract:** In European culture, only something could have names. Idea name-nothingness developed in India. The Arabic system, which is called the decimal system, originated in India. The digit “0” has a special place in our mind when it comes to numbers. The decimal representation of numbers differs from the standard way of writing and reading in European languages. This is because the decimal system was introduced to Europe by Muslim scholars. For instance, in Semitic languages such as Arabic, letters are written and read from right to left, and in European languages, the other way around, that is, from left to right.

**Keywords:** *zero, decimal representation of numbers, European vs Semitic ways of writing and reading*



# Spis tabel

Tabela I.1. Paradygmatyczne różnice między mechaniką klasyczną, mechaniką kwantową a koncepcją Rechnerer Raum.....	24
Table III.1. Patient characteristics.....	70
Table III.2. ICD-10 diagnosis codes statistics.....	71
Table III.3. Results using only the ICD-10 diagnosis codes present in the current episode, without patient history. The prefixes 3, 4, and 6 were used and then encoded with one hot. After encoding two kinds dimensionality reductions were performed using SVD to 200 and 50 dimensions. The learned classifier LGBM with parameter estimators 1000, max depth 3. ROC-AUC is shown with a 95% confidence interval .....	73
Table III.4. The results include patient diagnosis history for up to one year before admission .....	74
Table III.5. Results for all data, including current (prefix 4 no svd) and historical (prefix 6 and SVD 200) diagnoses .....	74





# Spis rysunków

Rysunek II.1. Przykład warstwy kompleksowej zbudowanej z $L$ neuronów formalnych $NF(\mathbf{w}_k, \theta_k)$ o liniowych regułach decyzyjnych (40).....	59
Figure III.1. The outcome of the episodes of care in 30 and 31–90 days after hospital discharge.....	69
Figure III.2. Patient age distribution .....	70
Figure III.3. Cross-validation loop with data transformation by an encoder. The encoder is fitted only on the training part of the data. The classifier is learned and evaluated on transformed data .....	72
Rysunek IV.1. Przedstawienie punktów w przestrzeni cech.....	80
Rysunek IV.2. Przedstawienie odwzorowania punktów z przestrzeni cech z Rys. IV.1 do przestrzeni parametrów .....	81
Rysunek IV.3. Funkcja kary .....	83
Rysunek IV.4. Jeden z dwuwymiarowych obrazów, na których testowany był algorytm mający na celu wykrywanie wzorców liniowych .....	84
Rysunek IV.5. Reprezentacja obrazu, która zostaje wykorzystana do dalszych obliczeń.....	84
Rysunek IV.6. Znalezione rozwiązania – wykryte zależności liniowe .....	85
Figure V.1. Block scheme of the proposed approach .....	92
Figure V.2. Sample scenario of testing an application – booking.com. Each segmented element is marked with a red square. After clicking on the selected element, the arrow will show the result.....	93
Figure V.3. Simple segmentation operations – original sample (a) and processed image (b).....	94
Figure V.4. Segmentation with k-Means – original image (a) and processed sample (b).....	94
Figure V.5. Scheme of the template matching algorithm used in the proposed framework .....	95
Figure V.6. Block diagram of the algorithm for low complexity images .....	96
Figure V.7. Examples of elements (template images) which are better handled by the feature-based algorithm (left) and which are better handled by the template-based algorithm (right).....	96

---

**lp** Politechnika  
Białostocka

