Konstruktywna indukcja cech we wspomaganiu decyzji na podstawie informacji obrazowej

Krzysztof Krawiec
Instytut Informatyki
Politechniki Poznańskiej

Praca doktorska
wykonana pod kierunkiem
prof. dr hab. inż. Romana Słowińskiego

Maj 2000

Oli, Faustynce i Dominice

Spis treści

Wstęp

W ostatnich latach bardzo popularne stało się stwierdzenie, iż współczesna cywilizacja opiera się w coraz większym stopniu na kulturze obrazu, która stopniowo wypiera kulturę słowa. Niezależnie od często negatywnych ocen tej tendencji, jest ona faktem i przybiera na sile, stanowiąc jeden z ważniejszych składników "cywilizacji multimedialnej". Trend ten należy przypisać głównie pojawieniu się w ostatnich dziesięcioleciach całej gamy nowych, niedostępnych do niedawna środków technicznych umożliwiających pozyskiwanie, przekazywanie i udostępnianie obrazów. Nowe formy komunikacji obrazowej spotykają się jednocześnie ze znacznym zainteresowaniem odbiorców, co uzasadnia się przede wszystkim tym, iż bodźce wizyjne w percepcji człowieka i większości innych ssaków odgrywają najważniejszą rolę. Szacuje się, że co najmniej 80% analizowanych przez mózg informacji pochodzi ze zmysłu wzroku [Ostrowski 1992], [Tomaszewski 1992], co znajduje odzwierciedlenie w stopniu zaawansowania i wielkości odpowiednich struktur w ośrodkowym układzie nerwowym.

Ta tendencja cywilizacyjna stanowi koło zamachowe zmian w naukach stosowanych. Zainteresowanie informacją obrazową w naukach technicznych, a w szczególności w informatyce i telekomunikacji jest znaczne, czego przejawem jest rosnąca popularność dziedzin realizujących **wspomaganie decyzji na podstawie informacji obrazowej (WDIO)**, takich jak przetwarzanie i analiza obrazów (ang. *image processing and analysis*) [Tadeusiewicz & Korohoda 1997], rozpoznawanie obrazów (ang. *pattern recognition*) [Tadeusiewicz & Flasiński 1991], [Gonzalez & Woods 1992], czy komputerowe widzenie (ang. *computer vision*) [Parker 1996], [Feldman & Bruckstein 1991]. Wiąże się to m.in. z dynamicznym rozwojem technik multimedialnych oraz szybkim rozprzestrzenianiem się Internetu i sieci telekomunikacyjnych o wysokiej przepustowości (por. np. [Stroiński & Węglarz 1997]).

Operując na dość wysokim poziomie ogólności metody przetwarzania i analizy informacji obrazowej można podzielić na (por. [Pavlidis 1987])

- *ilościowe*, tj. takie, które modyfikują jedynie zawartość obrazu, nie analizując go pod względem semantycznym i najczęściej nie zmieniając sposobu jego

reprezentacji (np. przetwarzanie obrazów rastrowych), oraz

- *jakościowe*, które ekstrahują z obrazu informację o wyższym stopniu abstrakcji, co najczęściej wymaga sięgnięcia do semantyki (np. interpretacji obiektów lub całego obrazu) i prowadzi do innej formy reprezentacji (np. rozpoznawanie obrazów).

Postęp dokonany w przeciągu kilku ostatnich dziesięcioleci w ilościowym przetwarzaniu informacji obrazowej jest niewątpliwie znaczny. Wypracowano całą gamę podejść i technik m.in. dla potrzeb akwizycji, przetwarzania, polepszania jakości, przesyłania i kompresji obrazów zarówno statycznych jak i dynamicznych (wideo). Natomiast metody jakościowej analizy obrazów, charakterystyczne dla rozpoznawania obrazów i widzenia komputerowego, nie doczekały się zbyt wielu spektakularnych osiągnięć. Przyczyn takiego stanu rzeczy należy upatrywać przede wszystkim w następujących czynnikach:

- stopniu komplikacji (i niekiedy złożoności obliczeniowej) zadania analizy i rozpoznawania obrazów,
- zróżnicowaniu rozwiązywanych problemów (przyczynia się do rozdrobnienia dziedziny na wiele autonomicznych nurtów badawczych),
- nadal niewystarczającym stanie wiedzy o sposobie przetwarzania bodźców wzrokowych w układzie nerwowym człowieka.

W konsekwencji takiego stanu rzeczy w analizie i rozpoznawaniu obrazów rozwijane są obecnie głównie *metody dedykowane do konkretnych zastosowań* (np. rozpoznawanie znaków alfanumerycznych, analiza zdjęć lotniczych i satelitarnych, analiza ruchu miejskiego). Wąska specjalizacja poszczególnych zastosowań utrudnia komunikację pomiędzy nimi i wypracowanie jednolitego modelu analizy informacji obrazowej. Projektowanie systemu wnioskującego z informacji obrazowej dla konkretnego zastosowania wymaga wiedzy dziedzinowej, której pozyskiwanie jest żmudne i kosztowne, jak pokazało już doświadczenie systemów eksperckich [Hayes–Roth, -Waterman, *et al.* 1983], [Jackson 1986], [Bubnicki 1990]. Wszystkie te problemy przyczyniają się do znacznej praco- i czasochłonności procesu projektowania systemu wnioskującego z informacji obrazowej i, w konsekwencji, jego znacznej kosztowności.

Niniejsza praca wychodzi naprzeciw potrzebie dokonania metodycznego uogólnienia. Jej zasadniczym **celem jest usprawnienie procesu projektowania systemu wspomagania decyzji na podstawie informacji obrazowej** (WDIO). W tym celu proponuje się metodę wnioskowania z informacji obrazowej, w której

wiedza o zadaniu (zastosowaniu) nabywana jest przez system w procesie uczenia, co uzyskuje się przez wykorzystanie metod charakterystycznych dla uczenia maszynowego. Głównymi cechami wyróżniającymi proponowane podejście spośród innych są:

- automatyczna konstrukcja cech obrazu (konstruktywna indukcja cech obrazu) odpowiednich dla danego zastosowania,
- ścisła integracja procesu uczenia z ekstrakcją cech,
- możliwość pozyskiwania metawiedzy, którą można wykorzystywać w nowych zastosowaniach.

Zaletą proponowanego podejścia jest przede wszystkim możliwość nauczenia się przez system rozwiązywania rozważanego problemu (zastosowania) wnioskowania z informacji obrazowej, co pociąga za sobą eliminację żmudnego etapu jawnego pozyskiwania wiedzy od eksperta i projektowania, stosownie do tej wiedzy, systemu wnioskującego z informacji obrazowej. Wykorzystanie uczenia daje też możliwość adaptacji zadań, których charakterystyka zmienia się w czasie.

Tematyka pracy i proponowane podejście mieszczą się zatem w nurcie jakościowej i semantycznej analizy informacji obrazowej. Wydaje się, że odpowiada to dobrze współczesnym trendom w rozwoju informatyki, która poza dotychczasowym zajmowaniem się głównie metodami gromadzenia, przechowywania, porządkowania i udostępniania danych i informacji, musi w obliczu szybkiego wzrostu ich objętości w coraz większym stopniu wspomagać człowieka w procesie ich analizy i wnioskowania [Słowiński 1992], [Słowiński 1998], [Węglarz 1998], [Vincke, Gassner, *et al.* 1992]. Dowodem na wzrost zainteresowania tym aspektem informatyki, nie tylko w środowiskach akademickich, ale także w domenie komercyjnej, może być dynamiczny rozwój takich dziedzin jak odkrywanie wiedzy (ang. *knowledge discovery*) i eksploracja danych (ang. *data mining*) [Fayyad, Piatetsky–Shapiro, *et al.* 1996]. Mamy zatem do czynienia z wyraźną reorientacją celów i zadań informatyki, która w coraz większym stopniu zajmuje się obecnie *wspomaganiem decyzji* (ang. *decision support*) człowieka (decydenta) w różnych problemach decyzyjnych, w miejsce podejmowania prób jego zastępowania [Roy 1990], [Słowiński 1997]. Jednocześnie wydaje się, iż w związku z coraz powszechniejszym wykorzystywaniem danych i informacji obrazowych i dźwiękowych, niezbędny jest rozwój *metod wspomagania decyzji bazujących na danych multimedialnych*. Podejście proponowane w niniejszej pracy może być w tym kontekście widziane jako pewna metodyka *wspomagania decyzji na podstawie informacji obrazowej*.

Poza opisem metodyki, niniejsza praca zawiera także opis jej zastosowania do rozpoznawania ręcznie pisanych znaków alfanumerycznych. Dla tych potrzeb powstała implementacja komputerowa wykorzystująca proponowane podejście.

Podziękowania

Większość eksperymentów obliczeniowych opisywanych w niniejszej pracy (rozdział 6) zostało przeprowadzonych na zbiorze obrazów znaków alfanumerycznych MNIST udostępnionych w Internecie przez **Y. LeCun'a** z AT&T Labs-Research [LeCun & et al. 1995], któremu składam podziękowanie za trud włożony w przygotowanie tego materiału.

Wyrazy wdzięczności wyrażam także prof. **Januszowi Szymasiowi** z Katedry Patomorfologii Klinicznej Akademii Medycznej w Poznaniu, który pozyskał i zdiagnozował obrazy nowotworów ośrodkowego układu nerwowego stanowiące przedmiot zastosowania prezentowanego w rozdziale 7.

Dziękuję dr **Jerzemu Stefanowskiemu** za cenne uwagi metodologiczne i terminologiczne, które pomogły mi w uporządkowaniu pracy.

Autor pragnie także wyrazić swoją wdzięczność **Komitetowi Badań Naukowych**, przy którego wsparciu (grant promotorski 8 T11C 021 16) powstała niniejsza praca. Rozprawa została przygotowana do druku z użyciem pakietu LaTeX i edytora Scientific Word^{TM}.

Lista ważniejszych symboli

Symbol	*Znaczenie*	*Strona*
U	zbiór wszystkich obiektów (obrazów)	29
x, x_i	obiekt (obraz), $x \in U$	29
C_k	k-ta klasa decyzyjna, $C_k \subset U$, $k \in \langle 1, n \rangle$	31
C^+	klasa (zbiór) przykładów pozytywnych[1]	32
C^-	klasa (zbiór) przykładów negatywnych	32
L	zbiór uczący, $L \subset U$	29
T	zbiór testujący, $T \subset U$	33
X	podzbiór przykładów, $X \subset U$	
X^+	podzbiór przykładów pozytywnych, $X^+ \subset X$	
X^-	podzbiór przykładów negatywnych, $X^- \subset X$	
η	trafność klasyfikowania	32
ϵ	błąd klasyfikowania	32
H	przestrzeń hipotez	31
h	hipoteza	30
f, f', f_j	cecha, $j \in \langle 1, m \rangle$	29
$\|f\|$	długość cechy f	62
$D^{-1}(f)$	zbiór wartości cechy	29
ϕ	wartość brakująca/nieznana (cechy)	30
F_0	oryginalny (początkowy) zbiór cech	29
F, F'	zbiór cech	29
F^*	optymalny zbiór cech	43
F_k	zbiór cech w k-tej iteracji procesu KI, $k > 0$	42
$F_{k^+} = F^+$	(suboptymalny) zbiór cech (wynik KI)	42
O	zbiór operatorów konstruktywnej indukcji cech	43
o_i	operator konstruktywnej indukcji cech, $o_i \in O$	43
$\digamma(F_0, O)$	przestrzeń reprezentacji (rozwiązań procesu KI)	46
E	funkcja oceniająca podzbiór cech, $D^{-1}(E) =< 0, 1 >$	43
A_0	zbiór cech atomowych	61
S	zbiór selektorów	64
A	zbiór agregatorów	65
$r(x)$	podstawowa reprezentacja obiektu (obrazu) x	61
a, a_i	składowa pierwotna obrazu, $a \in r(x)$	61
$R(x)$	pole widzenia w obrazie x	63

[1] dla dwuklasowego (binarnego) problemu uczenia maszynowego.

Rozdział 1

Cel i zakres pracy

1.1 Cel rozprawy

Niniejsza rozprawa dotyczy wykorzystania metod uczenia maszynowego do wspomagania decyzji na podstawie informacji obrazowej. W szczególności jej zasadniczym celem jest opracowanie, scharakteryzowanie i eksperymentalna weryfikacja nowej metody, która w procesie komputerowego wnioskowania z informacji obrazowej wykorzystuje konstruktywną indukcję cech.

1.2 Tło zagadnienia

Główną przesłanką dla postawienia powyższego celu badawczego było spostrzeżenie, iż mimo kilku dziesięcioleci intensywnego rozwoju metodologii przetwarzania i rozpoznawania obrazów, uczenie maszynowe i adaptacja zostały wykorzystane jedynie w niewielkim stopniu. Bezpośrednim następstwem takiej sytuacji jest mała elastyczność proponowanych metod i technik. W konsekwencji proces wnioskowania z informacji obrazowej musi być zazwyczaj dostosowywany indywidualnie do rozważanego zastosowania, co jest procesem żmudnym, słabo usystematyzowanym i kosztownym. Rozwiązania są tworzone zazwyczaj *ad hoc* dla konkretnych zastosowań i są trudne do przeniesienia lub uogólnienia na inne zastosowania.

Społeczność naukowa zajmująca się wnioskowaniem z informacji obrazowej ma świadomość tego problemu. Sądząc jednak po stosunkowo niewielkiej reprezentacji tego zagadnienia w literaturze, wydaje sie iż wielu badaczy pogodziło się do pewnego stopnia z brakiem ogólniejszej metodyki posiadającej cechy uczenia i adaptacji. Wskazywać na to może na przykład następujący cytat z monografii [Gonzalez & Woods 1992], będącej jednym z najbardziej popularnych podręczników przetwarza-

nia i analizy obrazu:

> *Despite this significant level of activity, the field [of image recognition and interpretation] remains a challenge. In particular, solutions of problems in image analysis are characterized by task-specific formulations, thus limiting the capability for advancement using the time-tried method of building a generalized body of results based on preceding accomplishments. For the foreseeable future, the design of image analysis systems will continue to require a mixture of art and science.* ([Gonzalez & Woods 1992], str.657)

Główna teza niniejszej rozprawy wynika z przeświadczenia autora o konieczności zakwestionowania prognozy zawartej w ostatnim z cytowanych zdań. Wraz z dynamicznym rozwojem technik informatycznych, rośnie liczba potencjalnych i już wdrożonych zastosowań przetwarzania i rozpoznawania obrazów. Wzrasta także stopień komplikacji problemów i proponowanych rozwiązań. W tym kontekście *niezbędne stało się wykorzystanie metod, które chociaż w części zautomatyzują proces projektowania i parametryzacji systemów wnioskujących z informacji obrazowej.* Niewątpliwie przydatne mogą tu być metody angażujące uczenie, oferując możliwość automatycznej (w pełni lub częściowo) adaptacji systemu wnioskującego z informacji obrazowej do specyfiki konkretnego zastosowania.

1.3 Zadania szczegółowe

Powyższy cel ogólny zamierzamy osiągnąć przez realizację następujących zadań szczegółowych:

- analiza możliwości automatycznej konstrukcji cech we wnioskowaniu z informacji obrazowej,
- opracowanie metody konstruktywnej indukcji cech z obrazów, a w szczególności zaproponowanie metod ekstrakcji cech składowych pierwotnych i operatorów konstruktywnej indukcji cech obrazu,
- zaprojektowanie i stworzenie implementacji komputerowej proponowanej metody,
- weryfikacja skuteczności proponowanej metody na praktycznych przykładach wnioskowania z informacji obrazowej.

Ponadto w ramach rozprawy realizowane są następujące podzadania o charakterze drugoplanowym:

- uporządkowanie i próba formalizacji zagadnienia konstrukcji cech,
- analiza ukierunkowania indukcyjnego konstruktywnej indukcji cech obrazu,
- ocena złożoności obliczeniowej proponowanego podejścia.

1.4 Cechy proponowanego podejścia

Głównymi cechami metody proponowanej w niniejszej pracy są:

- wykorzystanie **konstruktywnej indukcji cech do dynamicznego budowania cech obrazu w trakcie uczenia**,
- zdolność do samodzielnego wykształcania przez system uczący procedur selekcji **pola widzenia**, rozumianego jako część analizowanego obrazu,
- **ścisła integracja** uczenia z ekstrakcją cech (nie ograniczona jedynie do interfejsu pomiędzy modułem ekstrakcji cech a modułem uczącym, co jest podejściem najbardziej rozpowszechnionym (por. rozdział 3)).

1.5 Plan pracy

Organizacja niniejszej rozprawy jest podporządkowana realizacji powyższych zadań szczegółowych i przedstawia się następująco.

Rozdział 2 zawiera wprowadzenie w tematykę WDIO. Podane są w nim definicje pojęć wykorzystywanych w dalszych częściach rozprawy.

Rozdział 3 stanowi przegląd proponowanych metod WDIO wykorzystujących uczenie i adaptację. Podjęta jest w nim także próba dokonania klasyfikacji tych podejść ze względu na pewne cechy oraz scharakteryzowane są ich głowne zalety i wady.

Rozdział 4 stanowi wprowadzenie w uczenie maszynowe. W szczególności, w osobnym podrozdziale zaprezentowana jest konstruktywna indukcja cech.

Rozdział 5 zawiera opis proponowanego podejścia, a rozdziały 6 i 7 prezentują jego zastosowania w rozpoznawaniu ręcznie pisanych znaków alfanumerycznych i wspomaganiu diagnozowania nowotworów ośrodkowego układu nerwowego. Rozdział 8 zawiera wnioski wypływające z opisywanych testów w zastosowaniach praktycznych.

Dodatek A opisuje w szczegółach zestawy operatorów konstruktywnej indukcji użyte w poszczególnych eksperymentach. W dodatku B przedstawione zostały pewne aspekty implementacji komputerowej proponowanego podejścia, ze szczególnym uwzględnieniem usprawnień mających na celu skrócenie czasu obliczeń.

Uwagi terminologiczne

Mimo rosnącego od kilku lat zainteresowania uczeniem maszynowym w Polsce można chyba zaryzykowac twierdzenie, iż polskojęzyczna terminologia tej dziedziny nie wykształciła się jeszcze w pełni. Dlatego dla uniknięcia nieporozumień, w całej rozprawie przy definiowaniu pojęć zamieszczam w nawiasach ich anglojęzyczne odpowiedniki. Ponadto poniżej wyjaśniam kilka zasadniczych kwestii dotyczących terminów, których znaczenie może być szczególnie niejasne.

1. W niniejszej pracy pojawiają się często terminy **"wspomaganie decyzji na podstawie informacji obrazowej" (WDIO) i "wnioskowanie z informacji obrazowej", które zasadniczo odnoszą się do analizy i rozpoznawania obrazów, są jednak stosowane dla podkreślenia faktu ich rozpatrywania w kontekście wspomagania decyzji**.

2. W tradycji rozpoznawania obrazów (ang. *pattern recognition*) przez *obraz* rozumie się w ogólności pewien *wzorzec* (ang. *pattern*), który niekoniecznie musi być obrazem wizyjnym. Stąd wiele metod proponowanych w ramach tej dziedziny nadaje się do analizowania także sygnałów innych typów (np. przebiegów czasowych, map głębi, itp.). Niniejsza praca dotyczy jednak wyłącznie obrazów wizyjnych (informacji obrazowej).

3. Jako odpowiednik angielskiego terminu *machine learning* dla zwięzłości używam w pracy określenia *uczenie maszynowe*. Alternatywne propozycje *uczenie się maszyn* lub *maszynowe uczenie się* co prawda lepiej odzwierciedlają istotę tej dziedziny, wydają się jednak mniej naturalne.

4. Algorytm indukcji (w sensie uczenia maszynowego) określany jest niekiedy w rozprawie krótkim określeniem *induktor*, dla wyraźnego odróżnienia go od wygenerowanego klasyfikatora (patrz rozdział 4).

Rozdział 2

Analiza i rozpoznawanie obrazów

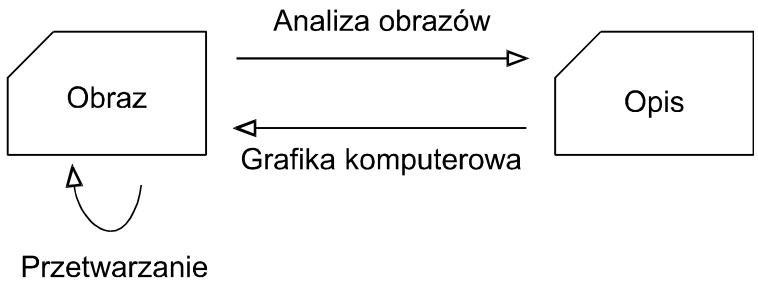
Rozdział ten stanowi krótkie wprowadzenie do **analizy i rozpoznawania obrazów, które z racji rozważania tej tematyki w kontekście wspomagania decyzji będzie dalej utożsamiane ze wspomaganiem decyzji na podstawie informacji obrazowej (WDIO)**. W ramach niniejszej pracy nie sposób dokonać kompletnej prezentacji tej dziedziny z racji jej obszerności. Dlatego rozdział ten ma miejscami charakter wybiórczy i prezentuje tylko te zagadnienia i pojęcia, które są niezbędne dla prezentacji zagadnień poruszanych w dalszych części pracy.

Charakterystyczną cechą WDIO jako dziedziny jest jej rozdrobnienie, przejawiające się w tym, że na poszczególnych aspektach WDIO koncentrują się względnie niezależne poddziedziny, jak na przykład przetwarzanie obrazów czy rozpoznawanie obrazów. Ta specjalizacja wynika m.in. z tego, że wnioskowanie z informacji obrazowej przebiega zazwyczaj w wielu etapach. Ponadto wykorzystywane podejście zależy w znacznym stopniu od postawionego celu, tj. od tego, czy interesuje nas klasyfikacja obrazów, rozpoznawanie obiektów w obrazach, przetwarzanie obrazów, itp. [Duda & Hart 1973].

2.1 Zadania WDIO

Metody wspomagania decyzji na podstawie informacji obrazowej operują zasadniczo na dwóch typach *danych*:

- obrazach (informacja o charakterze ilościowym i niskim stopniu uporządkowania),
- opisach obrazów (informacja o charakterze jakościowym i wysokim stopniu uporządkowania).



Rysunek 2.1: Wzajemne relacje poszczególnych specjalności względem obrazu i jego opisu

Poszczególne techniki stosowane we WDIO różnią się m.in. charakterem danych wejściowych i wyjściowych, co z kolei wynika z charakteru realizowanego zadania. Wokół poszczególnych zadań wykształciły się niezależne specjalności, z których do najważniejszych należą:

- przetwarzanie:
  - poprawa jakości (ang. *image enhancement*),
  - odtwarzanie obrazu (ang. *image restoration*),
- kompresja i kodowanie obrazu (ang. *image compression, image coding*),
- analiza obrazów
  - rozpoznawanie obrazów (ang. *pattern recognition*),
  - lokalizacja obiektów (ang. *object/target location*),
  - interpretacja (ang. *image interpretation, image understanding*),
  - komputerowe widzenie (ang. *computer vision, object recognition*, por. [Perrot & Hamey 1991]).

Umiejscowienie niektórych z tych specjalności względem obrazów i ich opisów ilustruje Rysunek 2.1 (za [Pavlidis 1987]). **Niniejsza praca jest poświęcona głównie zadaniu rozpoznawania obrazów**, jako jednemu z bardziej istotnych i najbardziej zbliżonemu do charakterystycznego dla uczenia maszynowego zagadnienia klasyfikacji.

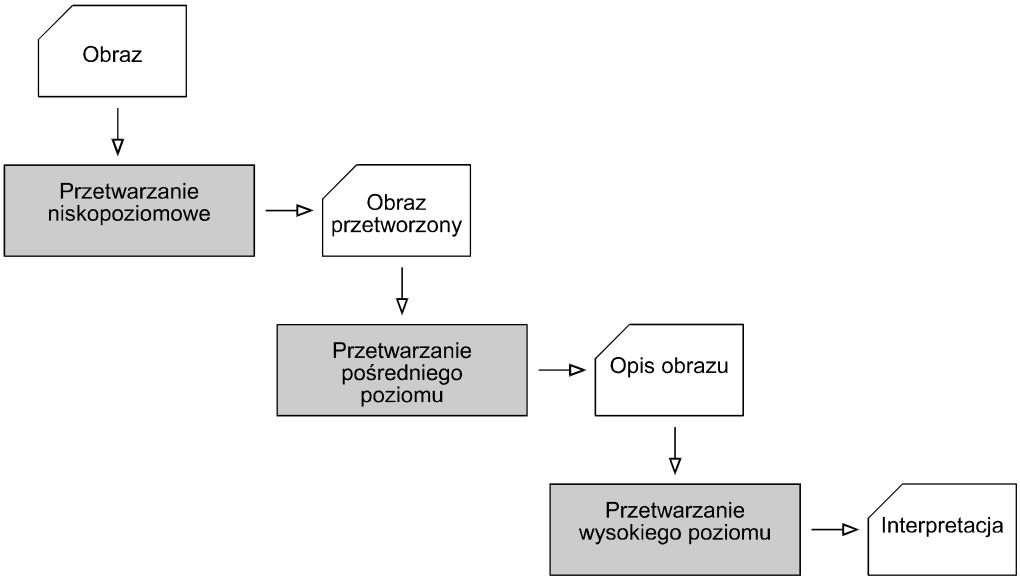
Należy także nadmienić, iż w związku z nasilającym się w ostatnich latach zainteresowaniem multimediami, coraz bardziej istotną rolę we WDIO odgrywają podejścia dedykowane do obrazów *dynamicznych*, które potocznie nazywa się obrazami wideo (ang. *video*). Specyfika tego zagadnienia polega m.in. na tym, iż tego typu obrazy muszą być w wielu zastosowaniach przetwarzane w czasie rzeczywistym (ang. *on-line*), co wymusza nacisk na ograniczanie złożoności obliczeniowej algorytmów używanych w tej klasie zastosowań. Niniejsza praca jest jednak poświęcona wyłącznie rozpoznawaniu obrazów *statycznych*, choć niewykluczona jest możliwość rozszerzenia proponowanego podejścia na klasę obrazów dynamicznych.

2.2 Etapy procesu WDIO

Tradycyjnie we wnioskowaniu z informacji obrazowej, a w szczególności w procesie rozpoznawania obrazu, prowadzącym od obrazu do pewnego wyniku jakościowego, wyróżnia się następujące etapy (por. np. [Gonzalez & Woods 1992]):

- przetwarzanie niskopoziomowe (ang. *low-level processing/vision*):
  - pozyskiwanie obrazu (ang. *image acquisition*),
  - wstępne przetwarzanie (ang. *image preprocessing*).
- przetwarzanie średniopoziomowe (pośredniego poziomu) (ang. *intermediate-level processing/vision*):
  - segmentacja (ang. *segmentation*),
  - generowanie opisu obrazu (ang. *description*).
- przetwarzanie wysokiego poziomu (ang. *high-level processing/vision*):
  - interpretacja (ang. *interpretation*),
  - rozpoznawanie (ang. *recognition*).

Sekwencję tych etapów przedstawia Rys. 2.2 (za [Gonzalez & Woods 1992], s. 573; por. też [Tadeusiewicz & Flasiński 1991], s. 133). Rysunek ten reprezentuje dość ogólny schemat, który obejmuje wiele zastosowań WDIO. Należy jednak jednocześnie nadmienić, iż w zależności od realizowanego zadania poszczególne etapy wnioskowania mogą być obecne lub nieobecne w procesie WDIO. Na przykład w polepszaniu obrazu WDIO kończy się w zasadzie na etapie wstępnego przetwarzania. Ponadto wiele z wyżej wymienionych terminów nie jest ścisłych i w literaturze



Rysunek 2.2: Etapy procesu wnioskowania z informacji obrazowej

przedmiotu nadal trwają spory o definicje poszczególnych procesów (np. *object recognition*, [Perrot & Hamey 1991], str. 1).

Jak widać z rysunku, konwencjonalny model WDIO bazuje na raczej jednokierunkowym przepływie informacji. Obserwacja ta ma kluczowe znaczenie dla niniejszej pracy.

2.3 Kryteria oceny systemów WDIO

Kryteria oceny systemów wnioskujących z informacji obrazowej można podzielić na dwie grupy:

- kryteria związane z *jakością* generowanych wyników, np. *trafność rozpoznawania* obrazu/obiektu, *dokładność lokalizacji* obiektu, etc.
- kryteria związane ze *złożonością* obliczeniową algorytmu.

Do podstawowych wymagań powiązanych z kryteriami jakościowymi charakteryzującymi metody WDIO, których spełnienie jest niezbędne w wielu zastosowaniach, należą:

- odporność na przesunięcia (ang. *translation, T*), skalowanie (ang. *scaling, S*) i obrót (ang. *rotation, R*) rozpoznawanych obrazów/obiektów,
- odporność na:
  - zaszumienie obrazu (np. w wyniku błędów transmisji),
  - zniekształcenie obrazu (np. w wyniku niedoskonałości toru optycznego),
  - niekompletność obrazu (np. w wyniku zasłonięcia pewnych obiektów innymi w obrazach scen trójwymiarowych),
- odporność na zmienność/różnorodność warunków akwizycji obrazu (np. oświetlenia sceny),
- stopień uniwersalności.

Istotność kryterium złożoności obliczeniowej zależy głównie od rozważanego zastosowania. Dla przeważającej większości metod i algorytmów kryteria związane ze złożonością są sprzeczne z kryteriami opisującymi jakość wyniku. Na przykład precyzyjna i odporna na zakłócenia procedura segmentacji obrazu przez rozrost obszaru jest zdecydowanie bardziej czasochłonna od szybkiej, ale bardzo "zgrubnej" segmentacji przez progowanie (por. np. [Pavlidis 1982]). Ta ortogonalność kryteriów jest jedną z przyczyn daleko posuniętej specjalizacji metod i algorytmów WDIO w zależności od zastosowania. Jest ona szczególnie widoczna na linii podziału pomiędzy zastosowaniami dla potrzeb obrazów statycznych i sygnału wideo. W niniejszej pracy, dotyczącej rozpoznawania obrazów, główny akcent położony jest na kryteria jakościowe, w szczególności na trafność rozpoznawania, odpowiadającą trafności klasyfikowania w nadzorowanym uczeniu maszynowym.

Rozdział 3

Uczenie w procesie WDIO

Rozdział ten rozpoczyna się od uzasadnienia potrzeby wykorzystania uczenia we wnioskowaniu z informacji obrazowej. Następnie przedstawiony jest przegląd literaturowy podejść wykorzystujących uczenie we WDIO połączony z próbą ich systematyzacji. W ostatnim podrozdziale dokonana jest krytyczna analiza niektórych z wymienionych wcześniej podejść oraz podane są argumenty przemawiające za potrzebą ściślejszej integracji uczenia z procesem WDIO.

3.1 Przesłanki dla wykorzystania uczenia we WDIO

Jak już zasygnalizowano w rozdziale 1, konwencjonalne systemy wnioskujące z informacji obrazowej zazwyczaj nie wykorzystują uczenia bądź wykorzystują je jedynie w nieznacznym stopniu. Pociąga to za sobą następujące wady:

- wysoki koszt i czasochłonność "ręcznej" implementacji wiedzy w systemie,
- brak lub małe możliwości przeniesienia bądź uogólnienia zaimplementowanej wiedzy na inne zastosowania,
- brak lub małe zdolności adaptacyjne (istotne dla problemów o zmiennej w czasie (dynamicznej) charakterystyce).

Z pierwszą z wymienionych wyżej wad związany jest ponadto problem *odpowiedniości reprezentacji wiedzy* eksperta z reprezentacją wiedzy w systemie WDIO. W wielu praktycznych zastosowaniach okazuje się na przykład, że cechy rozpoznawanych obrazów używane w praktyce przez ekspertów dziedziny zastosowania są

bardzo trudne do zaimplementowania w systemie WDIO, choć intuicyjnie wydają się zrozumiałe[1].

Coraz bardziej wyrafinowane i wymagające zastosowania praktycznie systemów wnioskujących z informacji obrazowej sprawiają, iż objętość wiedzy, którą należy zaimplementować w systemie, rozrasta się w znacznym tempie. Ponadto istnieje znaczne ryzyko pozostawienia w systemie pewnych luk, które mogą spowodować, iż w konfrontacji z sytuacją nietypową może on się zachować w nieprzewidziany sposób.

Potrzeba zaangażowania uczenia w proces WDIO stała się także szczególnie nagląca m.in. w związku z dynamicznym rozwojem robotyki. Robot nie tylko zbiera dane napływające ze świata zewnętrznego, ale także wchodzi z nim w pewne interakcje. Obserwując konsekwencje swych reakcji, może dokonywać ich oceny i poprawiać strategie wykonywania powierzonych mu zadań tak, aby w przyszłości realizować je lepiej. Takie postępowanie dokładnie odpowiada definicji uczenia (por. rozdział 4). Stąd w ostatnich latach zaobserwować można znaczne zainteresowanie *widzeniem aktywnym* (ang. *active vision*) w literaturze (zob. np. [Aloimonos 1993]).

Wyżej wymienione przesłanki prowadzą do ogólnej konkluzji o konieczności włączenia mechanizmów adaptacji i automatycznego pozyskiwania wiedzy o realizowanym zadaniu przez system WDIO. Uczenie, a w szczególności uczenie maszynowe, jest jednym z możliwych sposobów realizacji tego celu.

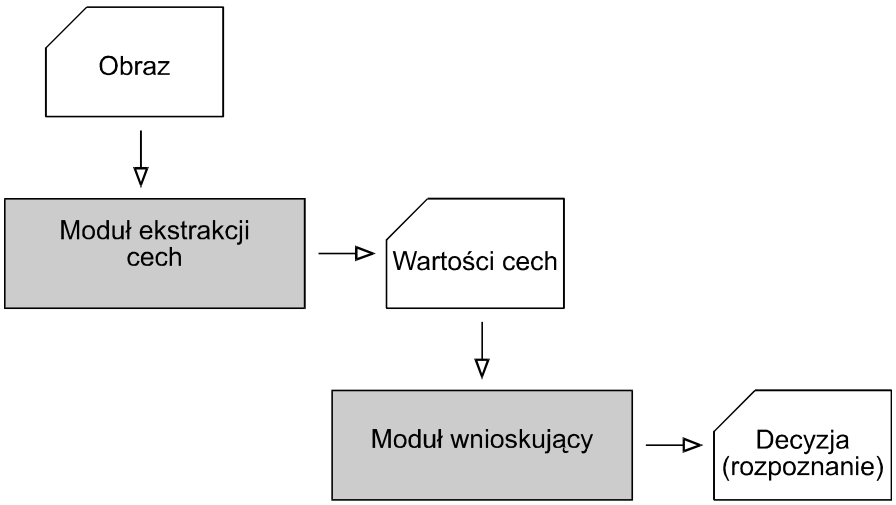
3.2 Próba systematyzacji podejść

W niniejszym podrozdziale dokonam próby uporządkowania (systematyzacji) podejść do WDIO wykorzystujących uczenie. W tym celu w kolejnych podrozdziałach zdefiniuję pewne cechy, które w istotny sposób różnicują te podejścia z metodologicznego punktu widzenia. Omówione zostaną także główne zalety i wady poszczególnych rozwiązań oraz zakres ich zastosowań.

Proponowana systematyzacja ma jednak charakter przybliżony, ponieważ ze względu na wspomnianą w rozdziale 2 różnorodność podejść obecnych we WDIO, trudno jest w wielu przypadkach wprowadzić klarowny podział metod ze względu na zastosowane poniżej kryteria[2].

[1] Przykładem może być tutaj termin *skóra tygrysia* stosowany w patomorfologii na określenie pewnego typu utkania komórkowego.

[2] Na przykład w wielu systemach WDIO trudno jest oddzielić reprezentację wiedzy dotyczącej rozpoznawanych obiektów (np. modelu) od meta-wiedzy dotyczącej strategii rozpoznawania.



Rysunek 3.1: Wnioskowanie w podejściu dwuetapowym

3.2.1 Sposób integracji uczenia z procesem WDIO

Podejścia wykorzystujące uczenie we WDIO należy przede wszystkim podzielić ze względu na *sposób, w jaki techniki uczenia zostały włączone do procesu WDIO*. Wyraźna linia podziału przebiega tutaj pomiędzy *podejściem dwuetapowym* a *podejściem z uczeniem zintegrowanym*. Ze względu na kluczowe znaczenie dla niniejszej pracy, rozróżnienie to zostanie dokładnie przedstawione w kolejnych podrozdziałach.

Podejście dwuetapowe

Zdecydowana większość spotykanych w literaturze propozycji traktuje wnioskowanie jako **proces niezależny od przetwarzania i analizy obrazu**. Dominującym rozwiązaniem jest podejście, które można scharakteryzować jako *dwuetapowe*, w którym przetwarzanie obrazu i wnioskowanie są realizowane przez dwa niezależne moduły. Ideę tego rozwiązania prezentuje Rys. 3.1 (por. np. [Tadeusiewicz & Flasiński 1991], s. 133).

Moduł pierwszy dokonuje zazwyczaj wstępnego przetwarzania obrazu i ekstrahuje z niego cechy istotne z punktu widzenia rozważanego zadania i zastosowania. Cechy te (czyli pewien *opis obrazu*) są następnie przekazywane do modułu wnioskującego.

Zadaniem modułu wnioskującego jest przeprowadzenie wnioskowania na podstawie dostarczonych cech. Do tego celu niezbędne jest, by posiadał on pewną *wiedzę* na temat rozważanego zastosowania. Wiedza ta może być pozyskana na co najmniej

dwa sposoby:

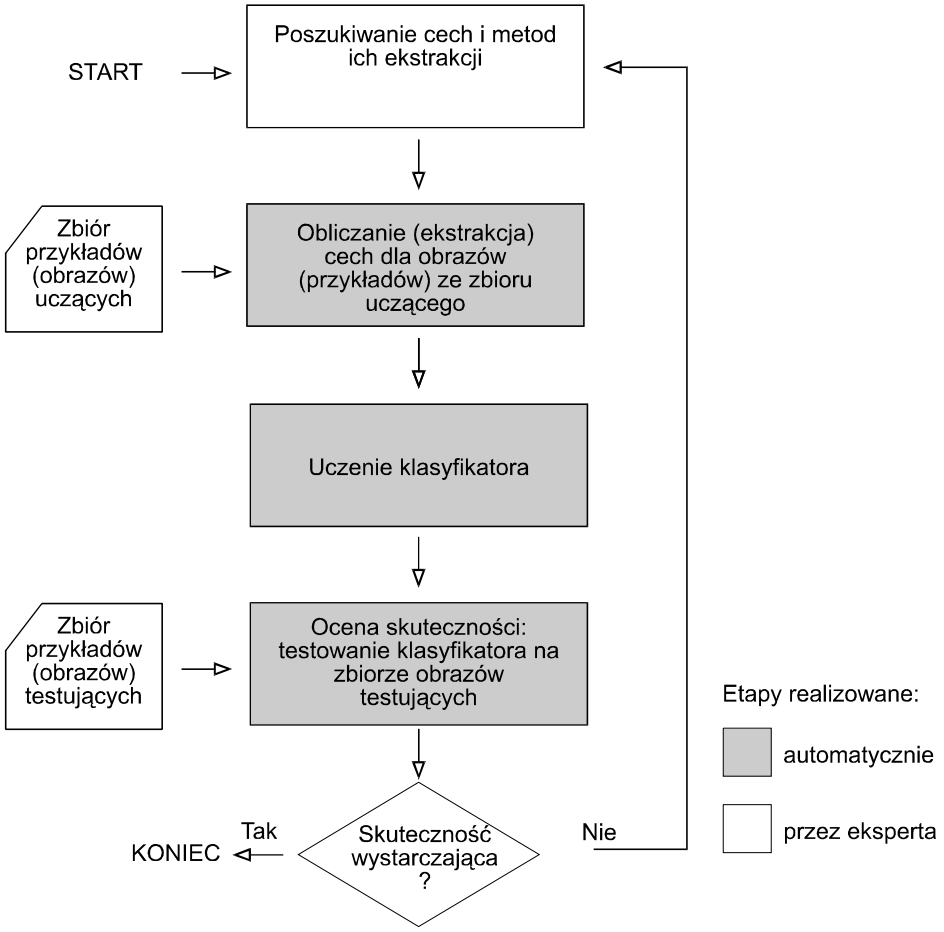
- jawnie (*explicite*), np. na drodze wywiadu (współpracy) z ekspertem dziedziny zastosowania, kompilacji wiedzy znanej z literatury przedmiotu, etc.
- automatycznie (lub pół-automatycznie), np. na podstawie danych opisujących przykładowe decyzje podejmowane w warunkach rozważanego zastosowania.

Pierwsze z wymienionych wyżej podejść jest charakterystyczne dla *systemów eksperckich* [Hayes–Roth, Waterman, *et al.* 1983], [Jackson 1986], [Mulawka 1996], [Bubnicki 1990]. Jego niewątpliwą zaletą jest pełna kontrola nad wiedzą zaimplementowaną w systemie i (przynajmniej do pewnych granic) możliwość wyjaśniania podejmowanych decyzji *w kategoriach znanych i charakterystycznych dla danej dziedziny.* Jawne pozyskiwanie wiedzy dla potrzeb systemu eksperckiego jest jednak procesem żmudnym i kosztownym, zwłaszcza dla nietrywialnych zastosowań, gdzie objętość wymaganej wiedzy jest znaczna. Jest to znane "wąskie gardło" systemów eksperckich (a także innych systemów, w których wiedza pozyskiwana jest w sposób jawny), które daje o sobie znać nie tylko w przypadku ich stosowania we WDIO (por. [Jackson 1986] rozdział 13, [Hayes–Roth, Waterman, *et al.* 1983]). Problem ten stał się jedną z zasadniczych przyczyn stopniowego spadku zainteresowania systemami eksperckimi w latach 90-tych.

W związku z powyższymi ograniczeniami, niniejsza rozprawa ogranicza się do rozważania drugiego z wymienionych wyżej rozwiązań, gdzie moduł wnioskujący pozyskuje wiedzę w sposób automatyczny. Jest to podejście bardzo szeroko wykorzystywane (por. np. [Jelonek, Krawiec, *et al.* 1994b], [Gonzalez & Woods 1992], [Karim & Alam 1998], [Alam & Karim 1998]) i sprawdza się w wielu zastosowaniach praktycznych. W szczególności, często korzysta się tu z dorobku *uczenia maszynowego*, któremu poświęcony jest w całości rozdział 4 niniejszej pracy.

Do niewątpliwych zalet podejścia dwuetapowego należą prostota koncepcyjna i łatwość implementacji (przy założeniu, że dany jest moduł ekstrakcji cech). Zalety te nie rekompensują jednak podstawowej wady tego podejścia, którą jest *brak wpływu modułu wnioskującego na działanie modułu przetwarzającego.* Sposób generowania opisu obrazu przez moduł przetwarzający nie podlega zmianom, w związku z czym uczenie w module wnioskującym przebiega w ustalonej *a priori* przestrzeni cech. Przestrzeń hipotez jakie mogą być wygenerowane jest rozpięta na cechach dostarczonych przez moduł przetwarzający, stopień swobody procesu uczenia jest zatem do tej przestrzeni ograniczony.

Ponadto należy pamiętać, iż w tej grupie podejść za dobór odpowiednich cech odpowiedzialny jest zazwyczaj projektant systemu WDIO. Bazując na swym doświadczeniu, wiedzy dziedzinowej i niekiedy intuicji, konstruuje on odpowiedni schemat



Rysunek 3.2: Proces projektowania systemu WDIO przy zastosowaniu podejścia dwuetapowego

wstępnego przetwarzania obrazu i dobiera taki sposób jego opisu (np. zestaw cech), po którym spodziewa się, że jest przydatne dla danego zastosowania (np. dobrze dyskryminuje klasy decyzyjne w problemie klasyfikacji). Jak już zaznaczono w rozdz. 1, jest to jednak proces trudny, w ogólności brak jest bowiem usystematyzowanej wiedzy o tym, kiedy i jak stosować poszczególne metody przetwarzania obrazów i ekstrakcji cech. W konsekwencji zazwyczaj nie mamy gwarancji znalezienia zbioru cech gwarantującego skuteczne rozpoznawanie, nie wspominając o takich charakterystykach jak minimalna liczność tego zbioru zapewniająca maksymalną trafność. Projektant systemu realizuje tu zatem trudniejszą część pracy, natomiast system uczący się jest w pewnym sensie "niedociążony" (patrz Rys. 3.2).

Sytuację tę dobrze charakteryzuje następujący cytat:

> *In many recognition systems (...), much time and man-power are usually spent in manual construction of class descriptions (...).* ([Nishida 1996], s. 400).

Podejścia z uczeniem zintegrowanym

W porównaniu z podejściem dwuetapowym, znacznie mniej licznie reprezentowaną w literaturze grupę stanowią podejścia, w których zachodzi wyraźne *przenikanie się procesów uczenia z przetwarzaniem obrazu*, lub inaczej, *podejścia z uczeniem zintegrowanym*.

W rozwiązaniach proponowanych w tej grupie ma miejsce zazwyczaj rozbudowa podejścia dwuetapowego o element *sprzężenia zwrotnego* od modułu wnioskującego do modułu przetwarzania obrazu. W takim ujęciu system uczący się nie jest już ograniczony do ustalonej przestrzeni poszukiwań, lecz może ją do pewnego stopnia modyfikować. Stąd rozwiązanie to nazywa się niekiedy *zamkniętą pętlą* (por. [Peng & Bhanu 1998], [Ahuja 1995]). W schemacie projektowania systemu WDIO prezentowanym na Rys. 3.2 odpowiada to (pełnej lub częściowej) automatyzacji etapów realizowanych dotąd przez eksperta, w tym zwłaszcza doboru cech.

Implementacja wyżej wymienionego sprzężenia zwrotnego może się odbywać m.in. poprzez (w kolejności od rozwiązań koncepcyjnie najprostszych do najbardziej zaawansowanych):

- dobór parametrów procedur przetwarzania obrazu,
- dobór procedur przetwarzania obrazu (oraz kolejności ich stosowania),
- dobór parametrów procedur generujących opis obrazu (np. ekstrahujących cechy) i ich parametryzację.

W innych, bardziej skomplikowanych podejściach, procesy uczenia mogą być wplecione bezpośrednio w przetwarzanie informacji obrazowej.

3.2.2 Reprezentacja obrazu

Rozpoznawanie zawsze polega na odniesieniu analizowanego obiektu do pewnego *wzorca* (*modelu*, *prototypu*), przy czym w zależności od podejścia porównanie to odbywa się w sposób mniej lub bardziej jawny. Metody rozpoznawania różnią się m.in. pod względem *reprezentacji*, jakiej używają w procesie rozpoznawania. W rozpoznawaniu obrazów wizyjnych charakterystyczna jest znaczna złożoność oryginalnej reprezentacji obrazu (np. mapy bitowej), stąd spotykane tu reprezentacje są zazwyczaj uproszczeniem obrazu oryginalnego. Dla prawidłowego funkcjonowania systemu niezbędne jest użycie reprezentacji, która nie ignoruje aspektów istotnych dla rozpoznawania. Poszczególne podejścia charakteryzują się różnym stopniem uproszczenia reprezentacji. Charakterystyczny jest tu pewien przetarg pomiędzy stopniem uproszczenia reprezentacji a stopniem komplikacji procedury rozpoznającej.

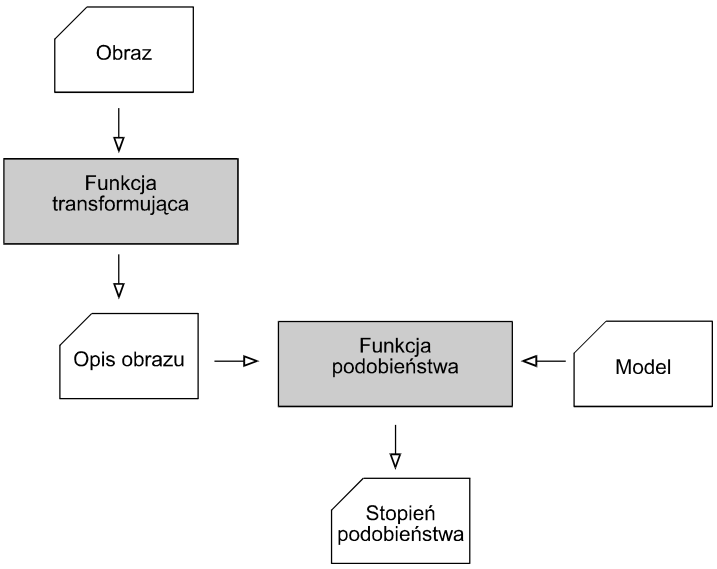
Podstawową cechą charakterystyczną obrazów wizyjnych jest ich *strukturalność* rozumiana najczęściej jako *względne lub bezwzględne umiejscowienie* poszczególnych obiektów w obrazie. Stąd ważny podział metod rozpoznawania przebiega właśnie po tej linii, dzieląc podejścia RO na (por. np. [Zabawa 1994]):

- *strukturalne* (używające reprezentacji obrazu uwzględniającej strukturę obrazów),
- *wektorowe* (używające reprezentacji obrazu ignorującej strukturę),
- *strukturalno-wektorowe*, używające reprezentacji hybrydowej.

Podejścia wektorowe biorą swą nazwę stąd, iż używaną przez nie reprezentacją jest *wektor* cech (skalarnych) opisujących obraz. W kolejnych podrozdziałach wymienione podejścia zostaną omówione dokładniej, wraz ze spotykanymi w literaturze metodami charakterystycznymi dla tych podejść.

Metody strukturalne

Podejścia oparte na modelu należą do dorobku tradycyjnego rozpoznawania obrazów (ang. *pattern recognition*). Zakłada się tu, że dla (jednej lub wielu) kategorii rozpoznawanych obiektów da się stworzyć pewien model (lub wiele modeli). W podejściach tych niezbędne jest zdefiniowanie trzech komponentów:



Rysunek 3.3: Wnioskowanie w metodach strukturalnych

- modelu (lub zbioru modeli),
- przekształcenia transformującego,
- miary podobieństwa.

Model jest uproszczoną, wyrażoną w pewnym języku, najczęściej strukturalną reprezentacją obiektów podlegających rozpoznawaniu.

Przekształcenie transformujące generuje opis obrazu w języku, w którym sformułowany jest model. Podobnie jak model, opis jest uproszczony w stosunku do oryginalnego obrazu.

Miara podobieństwa oblicza podobieństwo opisu obrazu do modelu. Podobieństwo to wyrażane jest często w kategoriach odległości. Zakłada się, iż podobieństwo opisu do modelu odzwierciedla stopień wiarygodności hipotezy o przynależności analizowanego obrazu do klasy decyzyjnej reprezentowanej przez model. Schemat podejścia opartego na modelu zilustrowany jest na Rys. 3.3. Warto zaznaczyć, iż podejście to jest w znacznym stopniu podobne do paradygmatu uczenia maszynowego zwanego *wnioskowaniem na podstawie zapamiętanych przypadków* (ang. *case-based reasoning, lazy learning*, por. [Aha 1991], [Weiss, Althoff, *et al.* 1994], [Kolodner 1993]).

Podejścia oparte na modelu są przydatne zwłaszcza w tych zastosowaniach, gdzie stosunkowo łatwo jest zdefiniować model dla rozpoznawanych obiektów. Ma to miej-

sce m.in. w niektórych zastosowaniach medycznych, robotyce i innych zastosowaniach przemysłowych.

W literaturze przedmiotu napotkać można wiele przykładów włączania technik uczenia do procesu rozpoznawania obrazów oparty na modelu i miarze podobieństwa. Podejścia te można podzielić na dwie grupy, w których uczenie zachodzi przez:

1. modyfikację miary podobieństwa, lub
2. modyfikację modelu.

Ad. 1. Prostszym i częściej spotykanym w literaturze jest pierwsze z wymienionych tu podejść. Jest ono najbardziej popularne w tych metodach, gdzie modele stanowią *strukturalną* reprezentację obrazu, tzn. są reprezentowane np. jako listy (ang. *strings*), drzewa (ang. *trees*), grafy (ang. *graphs*), itp. Podobieństwo opisu obrazu do modelu mierzy się zazwyczaj wówczas używając:

- odległości edycyjnych (ang. *edit distance*), lub
- miary podobieństwa lub dopasowania (ang. *matching*).

Odległość edycyjną definiuje się przez pewien zbiór elementarnych operacji modyfikujących wyrażenia przyjętego języka opisu (ang. *edit operations*). Wówczas odległość pomiędzy opisem obrazu a modelem definiuje się jako minimalną liczbę elementarnych operacji jakie należy zastosować, aby "przeprowadzić" pierwszy z nich w drugi. W bardziej wyrafinowanym wariancie do każdej elementarnej operacji przypisuje się pewną nieujemną wagę; w takim przypadku odległość jest równa najmniejszej możliwej sumie wag odpowiadających ciągowi operacji o wyżej wymienionej własności.

W odróżnieniu od wyżej wymienionej grupy, przy użyciu miar dopasowania nie dokonuje się modyfikacji opisu, lecz próbuje się odnaleźć wzajemną odpowiedniość poszczególnych elementów składowych. Miary te są szczególnie popularne dla reprezentacji grafowej. W szczególności, jeżeli możemy założyć że proces akwizycji obrazu oraz jego transformacja do przyjętego języka opisu nie wprowadzają istotnych zniekształceń, można oczekiwać, iż opis obrazu reprezentującego daną klasę decyzyjną będzie ściśle odpowiadał modelowi. W takim przypadku stosowane są algorytmy dokładnego dopasowania grafów (ang. *graph matching*). Funkcja podobieństwa ma wówczas charakterystykę binarną, gdzie zwracana wartość informuje jedynie o tym, czy udało się dopasować analizowane wyrażenia (grafy), czy też nie (czyli innymi słowy, czy istnieje izomorfizm pomiędzy analizowanymi grafami).

Założenie o braku zniekształceń opisu w stosunku do modelu jest jednak mało realistyczne z praktycznego punktu widzenia. Ponadto poważną wadą miar dopasowania jest fakt, iż dla wielu popularnych reprezentacji (grafy skierowane i nieskierowane) złożoność algorytmu testującego izomorfizm jest wykładnicza [Eshera & Fu 1984]. Stąd bardziej popularne jest rozważanie *podobieństwa* grafów (ang. *graph similarity measures, approximate graph matching*) zamiast poszukiwania ich dokładnego dopasowania. W tej grupie szczególnie ciekawe i popularne były m.in. podejścia wykorzystujące sztuczne sieci neuronowe. Na przykład w pracy [Bienenstock & von der Malsburg 1987] opisywane są różne warianty wykorzystania w tym celu sieci Hopfielda [Hopfield 1979]. Technika ta polega na tym, iż dla pary grafów, których podobieństwo chcemy zmierzyć konstruuje się sieć neuronową o wagach dobranych na podstawie ich struktury. Następnie pobudza się tak skonstuowaną sieć pewnym wymuszeniem i czeka na jej ustabilizowanie w pewnym punkcie równowagi. Wartość jaką osiąga funkcja energetyczna dla tego punktu jest miarą podobieństwa analizowanych grafów.

Ad. 2. W literaturze stosunkowo niewiele prac poświęcono WDIO opartym na modelu z uczeniem polegającym na modyfikacji modelu. Doniesienia, które napotkałem da się pogrupować w dwie kategorie:

- "dostrajanie" modelu danego a priori.
- pozyskiwanie modelu z danych (przykładów),

Pierwsze z wymienionych podejść jest niewątpliwie prostsze w sensie koncepcyjnym. Zakłada się tu, iż model rozpoznawanego obiektu jest dany (np. dostarczony przez eksperta dziedziny zastosowania). Może być on jednak niedokładny lub niekompletny. W trakcie uczenia model modyfikowany jest tak, aby jak najlepiej odzwierciedlać rozpoznawane obiekty.

Automatyczne pozyskiwanie *całego* modelu obiektu od podstaw jest zadaniem zdecydowanie trudniejszym. Liczba stopni swobody algorytmu uczenia jest tu nieporównywalnie większa niż w poprzednim przypadku. W ogólności należy liczyć się tu zatem z koniecznością dostarczenia systemowi uczącemu się zdecydowanie większej liczby przykładów.

Prace nad automatycznym pozyskiwaniem modelu z przykładów są intensywnie prowadzone w ramach *syntaktycznego rozpoznawania obrazów*. W kontekście tej gałęzi rozpoznawania obrazów, uczenie się modelu jest równoważne uczeniu się gramatyki formalnej (głównie jej produkcji) na podstawie przykładów słów należących do języka przez nią wygenerowanego. Zagadnienie to było w szczególności także popularne w Polsce [Flasiński 1992], [Flasiński 1991], rozdziały 9-12 w [Tadeusiewicz & Flasiński 1991]. Ostatecznie jednak proponowane rozwiązania nie znalazły zbyt

wielu zastosowań praktycznych, czemu na przeszkodzie stanęła wysoka złożoność obliczeniowa algorytmów uczących i algorytmów parsingu, mała odporność rozpoznawania na zakłócenia i zniekształcenia analizowanych obrazów oraz małe zdolności generalizowania opisu, szczególnie ważne z punktu widzenia uczenia maszynowego.

Metody wektorowe

Metody wektorowe są szczególnie popularne w podejściu dwuetapowym do rozpoznawania (patrz podrozdział 3.2.1). Obraz opisywany jest wówczas zazwyczaj za pomocą ustalonego przez projektanta zestawu (wektora) cech. Cechy te są następnie wykorzystywane przez moduł wnioskujący. Przydatność opisu wektorowego jest szczególnie wysoka w sytuacji, gdy trudno jest zbudować model rozpoznawanych obiektów (obrazów), co na przykład ma miejsce, gdy obrazy mają mało strukturalną charakterystykę.

Rozpoznawanie w oparciu o cechy obiektu w połączeniu z podejściem dwuetapowym jest niewątpliwie najbardziej popularnym i najczęściej spotykanym w literaturze schematem rozpoznawania. Z pewnym przybliżeniem można je określić inaczej jako rozpoznawanie *sterowane danymi* (ang. *data driven*). Warto podkreślić, że ten sposób rozpoznawania jest najbardziej dogodny dla stosowania konwencjonalnych algorytmów uczenia maszynowego w budowie modułu wnioskującego (por. rozdział 4).

3.2.3 Reprezentacja wiedzy

Istotne znaczenie we wnioskowaniu z informacji obrazowej ma forma (język) reprezentacji wiedzy o rozważanym problemie. Podobnie jak w sztucznej inteligencji, wykształcił tu się wyraźny podział na metody używające *symbolicznej* i *podsymbolicznej* (ang. *subsymbolic*, rozproszonej, ang. *distributed*) reprezentacji wiedzy.

W ramach WDIO **symboliczna** reprezentacja wiedzy wykorzystywana jest głównie do przechowywania modeli obiektów wykorzystywanych w metodach strukturalnych (por. punkt 3.2.2). Poza tym takiej reprezentacji wiedzy może używać klasyfikator implementujący moduł wnioskujący w podejściu dwuetapowym, jednak jest to wówczas bardziej cecha charakterystyczna wykorzystanej metody uczenia maszynowego (rozdział 4), niż całego systemu WDIO. Raczej rzadkie są przypadki wykorzystywania symbolicznej reprezentacji wiedzy do *bezpośredniego* wnioskowania z informacji obrazowej.

Podsymboliczną reprezentację wiedzy wykorzystuje model tzw. *obliczeń neuronowych*, znany potocznie pod nazwą *sztucznych sieci neuronowych* (SSN, ang. *artificial neural networks*) [Rumelhart, McClelland, *et al.* 1986], [Tadeusiewicz 1993],

[Tadeusiewicz 1993], [Hertz, Krogh, *et al.* 1993], [Rutkowski 1996], który począwszy od połowy lat osiemdziesiątych zdobył znaczną popularność w rozwiązywaniu problemów decyzyjnych i optymalizacyjnych. W szczególności także w ramach uczenia maszynowego zwrócono uwagę na przydatność tego podejścia m.in. do problemu klasyfikacji, w tym w otoczeniu nadzorowanym. Poza pewnym dorobkiem metodologicznym, w literaturze pojawiła się znaczna liczba prac opisujących praktyczne zastosowania SSN (por. np. przeglądy w [Żurada, Barski, *et al.* 1996], [Widrow, Rumelhart, *et al.* 1994]). W wielu przypadkach SSN wykazywały się lepszą zdolnością uogólniania.

Osiągnięcia SSN nie uszły także uwagi badaczom z kręgu WDIO. Wiele fundamentalnych prac z zakresu SSN dotyczyło matematycznego modelowania ośrodków wzroku [Grossberg 1976], [Grossberg 1976] i rozpoznawania obrazów (por. np. prace Hopfielda i Hintona w [Rumelhart, McClelland, *et al.* 1986], prace Fukushimy [Fukushima 1980], [Fukushima 1975]). Ponadto dla wybranych zastosowań udało się stworzyć oryginalne i niekiedy niezwykle skuteczne rozwiązania oparte o SSN (patrz punkt 3.3). Jednak można chyba stwierdzić, iż w ostatnim czasie doszło do pewnego spadku zainteresowania SSN jako metodą WDIO. Przyczyn tego stanu rzeczy można doszukiwać się m.in. w tym, że większość zastosowań SSN do WDIO reprezentuje podejście *bezpośrednie*, starając się do pewnego stopnia naśladować działanie receptorów wzrokowych. W ramach tego podejścia zakłada się, że na pojedyncze wejście (akson) sieci podawana jest wartość (najczęściej jasność) pojedynczego punktu obrazu (ang. *pixel*) lub niewielkiego lokalnego otoczenia. W konsekwencji architektura sieci silnie zależy od rozmiaru prezentowanych obrazów. Dlatego tak skonstruowane sieci da się stosować jedynie

- do obrazów o niewielkich rozmiarach/rozdzielczości (np. znaki alfanumeryczne),
- w zastosowaniach, gdzie rozpoznawane obrazy nie podlegają znacznym przesunięciom, skalowaniu i obrotom (por. punkt 2.3).

3.2.4 Charakter procesu uczenia

W istniejących metodach WDIO angażujących uczenie można dokonać także podziału ze względu na charakter zmian, jakie zachodzą w systemie w trakcie uczenia. W szczególności można wyróżnić pewną klasę systemów, w których w trakcie uczenia zachodzą jedynie zmiany *ilościowe* (np. dobór parametrów procedur przetwarzających obraz). W dalszej części pracy określam je mianem systemów *adaptacyjnych*. Drugą klasę tworzą systemy, w których w trakcie uczenia zachodzą zmiany *jakościowe* (np. zmiana struktury modelu reprezentującego rozpoznawany obiekt). Choć jest to podział raczej nieformalny, ponieważ granica pomiędzy zmianami ilościowymi

i jakościowymi jest raczej umowna, to jednak może on pomóc w uporządkowaniu istniejących podejść.

3.3 Przegląd wybranych podejść

W niniejszym podrozdziale dokonam przeglądu wybranych podejść wykorzystujących uczenie we WDIO wraz z ich zastosowaniami. W literaturze znaleźć można wiele pozycji przeglądowych dotyczących metod rozpoznawania obrazów i obiektów (np. [Perrot & Hamey 1991], [Parker 1996], [Fisher 2000]), jednak zazwyczaj nie biorą one pod uwagę aspektu uczenia. Do poniższego przeglądu wybrane zostały głownie te prace, w których uczenie zintegrowane jest w nowatorski sposób i/lub sprawdziły się one w jakimś zastosowaniu praktycznym.

Praca [Jain & Karu 1996] jest sztandarowym przykładem wykorzystania sztucznych sieci neuronowych w podejściu bezpośrednim. Rozważanym zastosowaniem było rozpoznawanie tekstur. Wykorzystano sieć neuronową o specjalizowanej architekturze, w której każdy neuron pierwszej warstwy zbiera informacje o intensywności poszczególnych punktów obrazu w niewielkim otoczeniu (polu widzenia, ang. *reception field*). Wyjścia sieci odpowiadają różnym klasom rozpoznawanych tekstur. Sieć uczona jest konwencjonalnym algorytmem wstecznej propagacji błędu [Rumelhart, McClelland, *et al.* 1986] na przykładach w postaci par *(obraz,klasa_tekstury)*. Autorzy ilustrują podejście dobrymi rezultatami w dwóch popularnych zastosowaniach: segmentacji dokumentów (ang. *page layout segmentation*) oraz w lokalizowaniu kodów paskowych (ang. *barcode localization*).

Literatura przedmiotu obfituje w podobne implementacje podejścia bezpośredniego z użyciem sieci neuronowych. Do najbardziej znanych z nich należą:

- system ALVINN wykorzystujący nauczoną sieć neuronową analizującą obraz z kamery do prowadzenia pojazdu [Pomerlau 1989]
- specjalizowane sieci neuronowe do rozpoznawania ręcznie pisanych znaków alfanumerycznych (*off-line*): neocognitron [Fukushima 1975], [Wake 1991], sieci LeCun'a [LeCun & Bengio 1994], [LeCun & Bengio 1995], także IRNN [Cios, Pedrycz, *et al.* 1998],
- system SKICAT do automatycznej identyfikacji, pomiaru i klasyfikacji obserwacji astronomicznych (ang. *Digital Palomar Observatory Sky Survery, POSS-II,* [Fayyad, Djorgovski, *et al.* 1996]),
- system PAPNET do wstępnej selekcji pól obserwacyjnych potencjalnie zawierających nieprawidłowe komórki w preparatach cytologicznych [Mango 1994].

W pracach [Draper 1993] i [Peng & Bhanu 1998] wykorzystano w ciekawy sposób uczenie ze wzmacnianiem (ang. *reinforcement learning* [Watkins 1989]) do "dostrajania" parametrów procesu rozpoznawania. System uczy się dobierać wartości parametrów algorytmu segmentacji na podstawie informacji zwrotnej o skuteczności rozpoznawania, czyli tzw. sygnału uczącego. Na przykład w przypadku pracy [Draper 1993], dotyczącej lokalizowania dachów budynków w zdjęciach lotniczych, sygnał uczący jest funkcją zgodności lokalizacji budynku wypracowanej przez system z lokalizacją rzeczywistą. Jak pokazano w rozdziale 2, wnioskowanie z informacji obrazowej jest ze swojej natury procesem wieloetapowym, gdzie informacja o trafności/skuteczności realizowanych kroków/etapów jest dostępna zazwyczaj dopiero na końcu procesu rozpoznawania. Sytuacja ta odpowiada dobrze paradygmatowi uczenia z opóźnioną nagrodą/wypłatą (ang. *learning with delayed reward*). W takim ujęciu reprezentacje analizowanego obrazu otrzymywane w kolejnych etapach przetwarzania odpowiadają stanom, natomiast poszczególne procedury przetwarzające - tranzycjom przeprowadzającym obraz (jego reprezentację) ze stanu w stan. Wynik otrzymany po ostatnim etapie przetwarzania oceniany jest przy pomocy odpowiednio skonstruowanej funkcji wypłaty, która odzwierciedla jego jakość (tu: zgodność rzeczywistej lokalizacji budynku z położeniem wskazanym przez system). Wadą proponowanego podejścia jest znaczna złożoność obliczeniowa (uczenie ze wzmacnianiem wymaga bardzo wielu iteracji dla zapewnienia zbieżności).

Prace [Maloof & Michalski 1997] i [Michalski, Rosenfeld, *et al.* 1997] opisują jedno z nielicznych rozwiązań, w których w kontekście rozpoznawania obrazów pojawia się zagadnienie konstruktywnej indukcji cech. Celem prac było wykrywanie zapalników ładunków wybuchowych w zdjęciach rentgenowskich bagażu lotniczego. Zaproponowano podejście dwuetapowe; w pierwszej fazie dokonuje się ekstrakcji z obrazu obiektów/pól widzenia (ROI) z użyciem konwencjonalnych technik przetwarzania obrazu (filtrowanie splotowe i progowanie). Następnie z każdego obiektu ekstrahuje się kilka elementarnych cech, m.in. obwód, pole powierzchni, zwartość (ang. *compactness*). Obiekty opisane cechami i przydzielone do jednej z dwóch klas decyzyjnych $\{zapalnik, nie\_zapalnik\}$ stanowią następnie zbiór uczący dla różnych algorytmów uczenia maszynowego. W pracach wykazano wyższość algorytmu AQ15c [Wnek, Kaufman, *et al.* 1995] nad konwencjonalnymi metodami uczenia maszynowego (klasyfikator minimalnoodległościowy (1NN), algorytm generowania drzew decyzyjnych C4.5 i sztuczna sieć neuronowa). Autorzy przypisują ten wynik większej elastyczności algorytmu AQ15c, wykorzystującego elementy konstruktywnej indukcji cech i rozbudowany język reprezentacji VL_1.

Praca [Francois & Medioni 1996] opisuje metodę angażującą wnioskowanie z zapamiętanych przypadków (*case-based reasoning*, CBR) do analizy i rozpoznawania kształtu. Po wstępnym przetwarzaniu kształt reprezentowany jest w postaci hierar-

chicznego grafu (drzewa), w którym węzły odpowiadają pewnym elementom strukturalnym, charakterystycznym dla teorii *geonów* Biedermana [Biederman 1985]. Elementy strukturalne charakteryzowane są przez typ symetrii krawędzi (równoległe/nie równoległe), krzywiznę (dodatnia, ujemna lub zero) oraz typ zakończenia (*mono-angular* i *pluri-angular*). Proces rozpoznawania ma miejsce w tak zdefiniowanej przestrzeni reprezentacji grafowych przy użyciu specjalnie zaprojektowanej miary podobieństwa, biorącej pod uwagę zarówno podobieństwo poszczególnych elementów strukturalnych (poprzez macierze kosztów) jak i podobieństwo struktury. Podobieństwo struktury obliczane jest poprzez algorytm przybliżony (o złożoności wielomianowej) działający w sposób hierarchiczny, od korzeni porównywanych reprezentacji ku liściom, czyli od reprezentacji najbardziej "zgrubnej" ku bardziej "detalicznej". Charakterystycznie dla CBR, autorzy proponują ponadto dodatkowy mechanizm hierarchicznego indeksowania zapamiętanych przypadków (grafów) w celu zminimalizowania czasu odszukania przykładu najbardziej podobnego do rozpoznawanego, oraz metodę szybkiego wstawiania nowego przypadku do tego indeksu (co odpowiada procesowi uczenia w CBR). Daje to możliwość użycia podejścia do problemów wymagających zapamiętania bardzo dużej liczby przykładów (prototypów), co jest jego poważną zaletą. Z drugiej strony należy podkreślić, że, charakterystycznie dla CBR, uczenie ogranicza się tu jedynie do umiejętnego wstawiania przykładów ze zbioru uczącego do bazy danych i nie obejmuje modyfikowania modelu lub procedury rozpoznającej.

W miarę podobne podejście prezentowane jest w pracy [Segen 1994], gdzie opisywany jest system GEST rozpoznający dwuwymiarowe kształty i określający ich położenie w polu widzenia. Kształty reprezentowane są w postaci grafu, w którym wierzchołkom odpowiadają tzw. cechy lokalne (punkty o wysokiej krzywiźnie), zaś krawędziom - relacje pomiędzy nimi. Uczenie na tak przetworzonych obrazach przebiega tu w kilku etapach: 1) nienadzorowana identyfikacja cech strukturalnych opartych na binarnych relacjach pomiędzy cechami lokalnymi, 2) nadzorowane uczenie, w wyniku którego powstają modele grafów reprezentujących poszczególne klasy decyzyjne, oraz 3) uczenie się modelu położenia (pozycji) rozpoznawanego kształtu w polu widzenia. System GEST skonstruowany został z myślą automatycznego rozpoznawania gestów w języku migowym, które to zadanie w eksperymentach opisywanych w cytowanej pracy wykonywał z ponad 90% trafnością.

Interesujące podejście oparte na modelu opisuje praca [Ristad & Yianilos 1998], gdzie proces uczenia dotyczy *sposobu* obliczania odległości (podobieństwa) pomiędzy obrazem x sprowadzonym do pewnej reprezentacji a modelem y wyrażonym w tych samych kategoriach. Rozważana jest tam reprezentacja obrazów w postaci łańcuchów (ang. *strings*), a do pomiaru ich podobieństwa wykorzystywana jest ciekawa miara zwana odległością edycyjną (ang. *string edit distance*). Miara ta jest w

ogólności pewną funkcją ciągu elementarnych operacji (np. usunięcia lub dodania elementu), jakie trzeba zastosować do obrazu (reprezentacji) x, aby przeprowadzić go w model y. Istotnym przyczynkiem cytowanej pracy jest propozycja automatycznego ustalania sposobu obliczania odległości na podstawie ciągu tak rozumianych operacji. W podobnym nurcie plasują się pionierskie prace Goldfarba [Goldfarb 1990], [Goldfarb 1992], [Goldfarb, Goldfarb, *et al.* 1995].

3.4 Podsumowanie

Prezentowany wyżej przegląd podejść do WDIO wykorzystujących uczenie może sprawiać wrażenie, że zagadnienie to jest dobrze poznane i naukowo "wyeksploatowane". Należy jednak zaznaczyć, iż rozkład prac reprezentujących poszczególne kierunki jest bardzo nierównomierny i prace prowadzone poza głównym nurtem podejść konwencjonalnych (dwuetapowych) są raczej nieliczne. Przyczyn takiego stanu rzeczy należy upatrywać głównie w stopniu komplikacji problemu, we wspomnianej w rozdziale 2 "fragmentacji" badań prowadzonych w ramach WDIO i, w konsekwencji, daleko posuniętej specjalizacji metod rozpoznawania i stosowanych reprezentacji obrazu, oraz w historycznym rozdziale dziedzin rozpoznawania obrazów i sztucznej inteligencji.

Podejście proponowane w niniejszej pracy (rozdział 5) wychodzi naprzeciw niektórym zasygnalizowanym wyżej problemom związanym z integracją uczenia z procesem WDIO, wykorzystując w tym celu pewne elementy uczenia maszynowego i konstruktywnej indukcji cech, opisywanych w następnym rozdziale.

Rozdział 4

Uczenie maszynowe i konstruktywna indukcja

Rozdział ten dokonuje krótkiego wprowadzenia w aparat pojęciowy uczenia maszynowego w zakresie wymaganym do przedstawienia proponowanego podejścia w kolejnym rozdziale. W szczególności, stosunkowo dużo miejsca poświęcam zagadnieniu ukierunkowania indukcyjnego oraz konstruktywnej indukcji cech, z racji ich wykorzystania w dalszej części rozprawy.

4.1 Geneza uczenia maszynowego

Uczenie maszynowe (ang. *machine learning*) to gałąź sztucznej inteligencji zajmująca się automatycznym *pozyskiwaniem wiedzy* z *danych* w celu wykorzystania jej w przyszłości [Michalski, Carbonell, *et al.* 1983], [Weiss & Kulikowski 1991], [Michalski, Bratko, *et al.* 1997], [Mitchell 1997], [Bolc & Zaremba 1992]. U źródeł uczenia maszynowego leżą m.in. negatywne doświadczenia z systemami eksperckimi, w których pozyskiwanie wiedzy odbywa się zazwyczaj przez dialog z ekspertem dziedziny zastosowania. W praktyce okazuje się, że jawna akwizycja obszernej bazy wiedzy jest zadaniem trudnym i czasochłonnym [Waterman 1986]. Natomiast obserwacje i opis *przykładów* konkretnych sytuacji decyzyjnych są z reguły dużo łatwiej dostępne i bardziej obiektywne.

Z podobnych przyczyn uczenie maszynowe staje się obecnie popularnym nurtem w ramach wielokryterialnego wspomagania decyzji (ang. *multicriteria decision support*). Tradycyjne metody charakterystyczne dla tej dziedziny budują (funkcyjny lub relacyjny) *model preferencji* w dialogu z decydentem, który musi zazwyczaj jawnie wyrażać swoje preferencje w kategoriach parametrów (wag, progów, współczynników wymiany, itp) przyjętego modelu (por. [Vincke, Gassner, *et al.* 1992]). Proces ten

często wymaga znajomości podstaw teoretycznych stosowanej metody wspomagania decyzji, co dla decydenta może być zadaniem trudnym. Stąd w ostatnich latach rośnie we wspomaganiu decyzji zainteresowanie metodami inferującymi preferncje decydenta z przykładów (przykładów wyboru, uporządkowania, sortowania, etc.). Podejście takie reprezentuje m.in. klasyczna już metoda UTA [Jacquet–Lagreze & Siskos 1982], gdzie na podstawie uporządkowania przez decydenta próby wariantów indukuje się model preferencji w postaci funkcji użyteczności. Do modelowania preferencji decydenta na podstawie ocen wybranych wariantów stosowano także specjalizowane sztuczne sieci neuronowe [Fraemling 1996]. Prawdziwie nowe podejście do modelowania preferencji, które z jednej strony wykorzystuje uczenie maszynowe, a z drugiej strony formę modelu preferencji w postaci wyrażeń logicznych typu reguł decyzyjnych wykorzystywanych do reprezentacji wiedzy w sztucznej inteligencji zaproponowali i scharakteryzowali Greco, Matarazzo i Słowiński [Greco, Matarazzo, *et al.* 1998], [Greco, Matarazzo, *et al.* 1999]. Podejście to oparte jest na odpowiednio dostosowanej koncepcji zbiorów przybliżonych [Pawlak 1982], [Pawlak 1991].

4.2 Podstawowe pojęcia

Do cech charakterystycznych uczenia maszynowego należą przede wszystkim:

- wykorzystanie metod i pojęć sztucznej inteligencji (np. symbolicznej reprezentacji wiedzy),
- uogólnianie (generalizacja) zdobytego doświadczenia w formie obserwacji lub przykładów decyzji,
- wyjaśnianie uzyskiwanych uogólnień i decyzji podejmowanych na ich podstawie.

Metody wykształcone przez uczenie maszynowe pozwalają budować inteligentne programy, które są zdolne do:

- pozyskiwania wiedzy o danym problemie (środowisku) na podstawie interakcji z nim,
- przetwarzania (zwłaszcza uogólniania) nabytej wiedzy w celu wykształcenia ogólnego modelu problemu,
- stosowania pozyskanej i przetworzonej wiedzy do podejmowania pewnych decyzji (akcji) w nowych sytuacjach decyzyjnych.

Uogólnianie pozyskanego doświadczenia na przypadki nie występujące w danych (zbiorze uczącym) jest centralnym zagadnieniem uczenia maszynowego. Stąd używa się często w tym kontekście terminu *indukcja*, przez co rozumie się generowanie na podstawie danych wiedzy bardziej ogólnej od tej, która wynika z nich poprzez wnioskowanie *dedukcyjne* (por. [Michalski, Bratko, *et al.* 1997], rozdział 1). Sposób uogólniania implementowany przez system uczący się wynika z realizowanego przezeń *ukierunkowania indukcyjnego*; zagadnieniu temu poświęcony jest jeden z następnych podrozdziałów.

4.2.1 Paradygmaty uczenia maszynowego

Ze względu na metodykę leżącą u podstaw poszczególnych technik uczenia, w uczeniu maszynowym można wyróżnić następujące główne paradygmaty, związane ze strategiami uczenia (por. [Bolc & Zaremba 1992]):

- uczenie się przez zapamiętywanie (ang. *case-based learning, rote learning*),
- uczenie się z instrukcji (ang. *learning by being told*),
- uczenie się przez indukcję (ang. *inductive learning*),
  - uczenie się z przykładów (ang. *learning from examples*),
- uczenie się przez dedukcję
- uczenie się przez analogię (ang. *learning by analogy*),
- uczenie się ze wzmacnianiem (ang. *reinforcement learning*),
- uczenie się z doświadczenia (ang. *learning from experience*),

Spośród tych paradygmatów najczęściej opisywanym w literaturze jest uczenie się z przykładów. Jego popularność wynika z faktu, iż charakterystyczne dla niego zadanie *klasyfikacji* stosunkowo najczęściej pojawia się w zastosowaniach. Ponadto jest ono znane i badane od co najmniej XIX wieku w naukach takich jak statystyka i taksonomia (por. np. [Pociecha, Podolec, *et al.* 1988]). Praca niniejsza koncentruje się właśnie na tym zagadnieniu.

4.2.2 Uczenie się z przykładów

Zadaniem systemu uczącego się z przykładów jest utworzenie ogólnego pojęcia na podstawie przykładów drogą klasyfikowania obiektów jako należących lub nienależących do poznawanej klasy (za [Bolc & Zaremba 1992]). Cechą charakterystyczną uczenia się z przykładów jest stosunkowo mały udział projektanta (np. eksperta dziedzinowego) w konstrukcji systemu uczącego się.

W ramach paradygmatu uczenia się z przykładów zakłada się, że *przykłady* (*obiekty*[1], ang. *examples, instances*) x_i odpowiadające *obserwacjom* w ujęciu statystycznym, pochodzą z pewnego uniwersum (populacji) U, która może być zbiorem skończonym bądź nieskończonym. Zadaniem systemu uczącego się jest pozyskanie wiedzy z pewnego podzbioru *przykładów* $L \subset U$, zwanego *zbiorem uczącym* (ang. *training set*), stanowiącego najczęściej losową próbę z populacji wszystkich obiektów U. Dla prawidłowego przebiegu procesu uczenia niezbędne jest, aby próba ta była odpowiednio liczna i *reprezentatywna*.

Obiekty opisane są pewnym zbiorem *cech* (atrybutów, zmiennych, ang. *feature, attribute*) $F = \{f_1, f_2, ..., f_m\}$ (rodzina funkcji), odpowiadających zmiennym w interpretacji statystycznej. Cecha jest *funkcją* odwzorowującą zbiór obiektów (przykładów) w zbiór wartości: $f_j : U \longrightarrow D^{-1}(f_j)$ [2]. Niech $f_j(x_i), f \in F$ oznacza wartość j-tej cechy dla przykładu x_i. Dogodnym sposobem prezentacji zbioru przykładów L jest postać *tablicy decyzyjnej* (zwanej także *systemem informacyjnym*), gdzie wiersze odpowiadają przykładom (obiektom) a kolumny - cechom. Wówczas w komórce tabeli znajdującej się na przecięciu i-tego wiersza i j-tej kolumny znajduje się wartość j-tej cechy dla i-tego przykładu.

Wartości cech są najczęściej "surowymi" lub przetworzonymi wynikami pewnych pomiarów lub są podane przez eksperta. Opisywana w dalszej części pracy (podrozdz. 4.4) konstruktywna indukcja cech (KI) prowadzi do powstawania nowych cech. Dlatego dla odróżnienia *cech oryginalnych* od cech generowanych w procesie KI, oznaczmy ten pierwszy przez F_0 .

Zbiór wartości (przeciwdziedzina) $D^{-1}(f_j)$cechy f_i zależy od jej interpretacji w dziedzinie zastosowania. Podobnie jak zmienne w statystyce, cechy można pogrupować stosownie do typów *skal*, na których są zdefiniowane. Do skal stosowanych powszechnie należą, w kolejności odzwierciedlającej wzrastający stopień uporządko-

[1] *Obiekt* jest zasadniczo dowolnym elementem uniwersum U, natomiast *przykład* to obiekt, dla którego znana jest jego przynależność do klasy decyzyjnej (patrz następne strony).

[2] W niniejszej rozprawie pojęcie cechy jest używane w znaczeniu funkcyjnym, tj. bierzemy pod uwagę *wartość* cechy dla danego obiektu. Warto jednak nadmienić, iż w literaturze rozpoznawania obrazów występuje ono także w innych znaczeniach, np. "obiekt x *posiada* cechę f". Jest to jednak szczególny przypadek cechy jako funkcji boolowskiej i w związku z tym nie pomniejsza on ogólności rozważań.

wania wartości: skala *nominalna*, skala *porządkowa*, skala *przedziałowa*, oraz skala *ilorazowa*. Skala cechy w znacznym stopniu determinuje możliwości stosowania poszczególnych technik uczenia maszynowego i konstruktywnej indukcji cech. Jednocześnie większość popularnych systemów uczenia maszynowego nie wprowadza jednak tak drobiazgowego rozróżnienia skal. Na przykład w sztucznych sieciach neuronowych traktuje się jednakowo cechy zdefiniowane na skalach porządkowej, przedziałowej i ilorazowej. Z kolei wiele algorytmów generowania reguł nie różnicuje pomiędzy skalą nominalną i porządkową (np. LEM2 [Chan & Grzymała–Busse 1994]).

W rzeczywistych problemach zdarza się niekiedy, iż opis przykładów w przestrzeni cech jest niepełny, co przejawia się w brakujących wartościach niektórych cech dla poszczególnych przykładów (ang. *missing/unknown attribute values*). Oznaczmy przez ϕ wartość brakującą. Zapis $f_j(x_i) = \phi$ oznaczać będzie brak wartości cechy f_j dla przykładu x_i.

m-wymiarową przestrzeń cech F nazywamy *przestrzenią reprezentacji* (ang. *representation space*) lub *przestrzenią przykładów* (ang. *example space*). Pojedynczy przykład (obiekt) z uniwersum U z określonymi wartościami wszystkich cech jest jednoznacznie reprezentowany przez *punkt* w tej przestrzeni.

W wyniku uczenia na zbiorze przykładów L opisanych cechami F system uczenia maszynowego generuje pewną *hipotezę* (opis) $h_F(L)$ (ang. *hypothesis*, *concept description*), lub krótko h_F. To właśnie hipoteza jest, ujmując rzecz precyzyjnie, *klasyfikatorem* (ang. *classifier*), w odróżnieniu od programu (algorytmu indukcyjnego), który ją wygenerował (ang. *inducer*). Potocznie jednak często stosuje się te pojęcia zamiennie, co może prowadzić do nieporozumień. Stąd w ramach niniejszej pracy na określenie metody uczenia maszynowego generującej klasyfikatory stosowany będzie termin *algorytm indukcji*, lub krótko *induktor*.

Hipoteza wyrażona jest w pewnym języku *reprezentacji wiedzy* używanej przez induktor. Języki (formy) reprezentacji wiedzy można podzielić na

- *symboliczne* (np. wyrażenia logiczne, reguły decyzyjne, drzewa decyzyjne, sieci semantyczne, ramy),
- *podsymboliczne* (ang. *subsymbolic*), reprezentowane głównie przez sztuczne sieci neuronowe.

Zazwyczaj język reprezentacji wiedzy narzucany jest przez induktor. Ponieważ kwestia ta nie ma większego znaczenia dla proponowanego podejścia, nie zostanie tu rozwinięta. Obszerne przeglądy algorytmów indukcji i języków reprezentacji wiedzy znaleźć można w literaturze [Bolc & Zaremba 1992], [Mitchell 1997].

Zbiór wszystkich hipotez H, które mogą być wygenerowane przez klasyfikator używający określonego języka reprezentacji wiedzy (najczęściej przy pewnych ograniczeniach narzuconych np. na długość hipotezy), nazywa się przestrzenią hipotez (ang. *hypothesis space*). Uczenie można zatem ująć jako problem przeszukiwania przestrzeni hipotez (ang. *space search problem*), którego celem jest odszukanie hipotezy "najlepszej" ze względu na pewne kryteria.

4.2.3 Uczenie nadzorowane i nienadzorowane

Metody uczenia maszynowego są zazwyczaj osadzone w jednym z dwóch *otoczeń*:

- otoczenie *nienadzorowane* (ang. *unsupervised setting/learning*),
- otoczenie *nadzorowane* (ang. *supervised setting/learning*).

W otoczeniu nienadzorowanym system uczący się generuje hipotezę jedynie w oparciu o wartości cech. Zadaniem systemu uczącego się jest wówczas wykrycie i opisanie pewnych regularności (prawidłowości, związków, porządku, skupień) w danych. Historycznie rzecz biorąc, jest to podejście charakterystyczne dla *taksonomii*; stosuje się tu takie metody jak analiza skupień, hierarchizacja dendrytowa, hierarchizacja drzewkowa, itp. (zob. np. [Grabiński, Wydymus, *et al.* 1989], [Pociecha, Podolec, *et al.* 1988]).

Z kolei w otoczeniu nadzorowanym, do którego ogranicza się niniejsza praca, zakłada się, że na zbiorze wszystkich obiektów U określony jest podział (*partycja*) $P(U)$ na *klasy decyzyjne* C_k. Podział ten, zgodnie definicją, spełnia warunki:

1. *pełności*

$$\bigcup_{k=1}^{n} C_k = U$$

2. *rozłączności* klas decyzyjnych

$$\forall\ C_k \subset U, C_l \subset U : C_k \cap C_l = \emptyset, k, l = \langle 1, n \rangle, k \neq l$$

gdzie n jest liczbą klas decyzyjnych. Podział na klasy decyzyjne wynika zazwyczaj z dziedziny zastosowania i dany jest a priori (np. przez eksperta z dziedziny zastosowania). Klasom decyzyjnym mogą na przykład odpowiadać rozpoznania w medycznych problemach diagnostycznych, decyzje finansowe w problemach ekonomicznych, rozpoznania obiektów we WDIO, itp. W otoczeniu nadzorowanym zadaniem systemu uczącego się jest wygenerowanie hipotezy h możliwie dobrze dyskryminującej klasy decyzyjne (patrz kolejny podrozdział); cechy nazywa się wówczas *atrybutami warunkowymi*.

Podział na klasy decyzyjne implementuje się często przez tzw. *atrybut decyzyjny*, będący zmienną nominalną lub porządkową. Atrybut decyzyjny nazywa się też niekiedy (np. w kontekście sztucznych sieci neuronowych) *sygnałem uczącym*, ponieważ to właśnie ta informacja steruje w znacznym stopniu wykształcaniem hipotezy w trakcie uczenia.

W ramach niniejszej pracy rozpatrywane są m.in. *binarne* problemy uczenia maszynowego, tj. zadania uczenia, gdzie $n = 2$. W takim przypadku rezygnuje się zazwyczaj z numerowania klas decyzyjnych, i w miejsce tego jedną z klas decyzyjnych nazywa się klasą pozytywną i oznacza przez C^+, drugą natomiast nazywa się klasą negatywną i oznacza przez C^-.

4.2.4 Weryfikacja systemu uczącego się

Zdolność predykcyjna

W otoczeniu nadzorowanym hipoteza $h_F(L)$ wygenerowana przez induktor w procesie uczenia służy do klasyfikowania obiektów. Hipoteza wyznacza w sposób jednoznaczny pewną partycję $P_h(U)$ na uniwersum obiektów. Wówczas miarę zgodności oryginalnego podziału na klasy decyzyjne z podziałem (klasyfikacją) wynikającym z działania klasyfikatora dla zbioru obiektów $X \subseteq U$ zdefiniowaną jako

$$\eta(h, X) = \frac{1}{|X|} \left| X \cap \bigcup_{k=1}^{n} C_k \cap C'_k \right|, \quad C_k \in P(U), C'_k \in P_h(U)$$

nazywamy *trafnością klasyfikowania* (ang. *accuracy of classification*) klasyfikatora h na zbiorze przykładów X. Komplementarną wielkość

$$\epsilon(h, X) = 1 - \eta(h, X)$$

nazywamy *błędem klasyfikowania* (ang. *classification error*) hipotezy (klasyfikatora) h na zbiorze przykładów X. Obie te wielkości przyjmują wartości z przedziału $\langle 0, 1 \rangle$ i są często dla czytelności wyrażane w procentach.

Gdyby celem klasyfikacji było jedynie możliwie skuteczne rozpoznawanie obiektów ze zbioru uczącego L, zadanie uczenia nadzorowanego można by ująć w kategoriach problemu optymalizacji. Jednocześnie byłoby możliwe rozwiązanie trywialne, ponieważ dla bezbłędnego klasyfikowania przykładów ze zbioru uczącego wystarczy je zapamiętać.

W rzeczywistości zbiór przykładów uczących L jest jednak pewną ograniczoną *próbą* wyjętą z populacji wszystkich przykładów U, która jest często nieskończenie duża. O przydatności danego klasyfikatora decyduje zaś jego zdolność *uogólniania*,

która pozwala mu klasyfikować poprawnie nie tylko przykłady występujące w zbiorze uczącym, ale także obiekty spoza niego ($U \backslash L$). Stąd błąd klasyfikowania dla zbioru uczącego $\epsilon(h, L)$ nazywa się często *błędem pozornym* (ang. *apparent error*), dla podkreślenia faktu, że w ogólności nie odzwierciedla on skuteczności klasyfikatora na całej populacji przykładów U. Na określenie niepożądanego zjawiska uzyskiwania przez klasyfikator wysokiej trafności klasyfikowania na zbiorze uczącym i niskiej dla obiektów spoza niego, czyli

$$\eta(h(L), L) > \eta(h(L), U \backslash L) \tag{4.1}$$

używa się terminu *przeuczenie* (ang. *overfitting*). Przeuczenie jest objawem zbyt dokładnego "dopasowania" klasyfikatora wygenerowanego przez induktor do zbioru uczącego i jej za mało ogólnego charakteru. Problem przeuczenia jest jednym z trudnych zagadnień uczenia maszynowego.

Uwzględniając powyższe spostrzeżenia, problem nadzorowanego uczenia się z przykładów należy zatem sformułować (przyjmując pewne uproszczenie polegające na pominięciu innych kryteriów oceny klasyfikatorów wymienionych w punkcie 4.2.4) jako poszukiwanie hipotezy maksymalizującej zdolność predykcyjną, co da się zapisać jako:

$$\max_{h \in H} z = \eta(h(X), U) \tag{4.2}$$

gdzie $X \subset U$ jest zbiorem dostępnych przykładów. Ponieważ jednak optymalizowanej tu wielkości nie da się obliczyć ze względu na brak dostępu do przykładów spoza X ($U \backslash X$), rozwiązywanie rzeczywistych problemów uczenia maszynowego sprowadza się w praktyce do optymalizacji pewnej estymaty $\eta(h(X), U)$.

W najprostszym przypadku jest to trafność klasyfikowania dla pewnego podzbioru $T \subset X$ zawierającego obiekty nie wykorzystywane do generowania klasyfikatora ($T \cap L = \emptyset$)[3], zwanego *zbiorem testującym*[4]. Wielkość $\eta(h(T), U)$ nazywa się często *zdolnością predykcyjną klasyfikatora* (ang. *prediction accuracy*) lub krótko *trafnością klasyfikowania*. W rzeczywistych zastosowaniach podział dostępnego zbioru obiektów X na zbiór uczący i testujący nie zawsze jest dany a priori. W takim przypadku przeprowadza się jego losowy podział na T i L (najczęściej z równomiernym rozkładem prawdopodobieństwa lub tak, aby zapewnić możliwie zgodne częstości występowania przykładów z poszczególnych klas decyzyjnych w zbiorze uczącym i testującym).

[3] Jest to tzw. *OTS*, *off-training set regime*; niektóre podejścia do uczenia się z przykładów dopuszczają wyjątki od zasady rozłączności zbioru uczącego i testującego, por. np. [Bensusan 1998].

[4] Zgodnie z uwagą na stopce strony 29, obiekty ze zbioru testującego nie są *przykładami*, gdyż (z punktu widzenia klasyfikatora) ich przynależność do klas decyzyjnych nie jest znana.

Zastosowanie jednokrotnego podziału zbioru X nie daje jednak zazwyczaj wiarygodnych wyników, ponieważ nawet dla prostych problemów uzyskana trafność klasyfikowania może się silnie zmieniać w zależności od tego, które przykłady w wyniku losowania trafiły do zbioru uczącego, a które do testującego. Dlatego, aby zapewnić bardziej pewną estymatę zdolności predykcyjnej, przydatne jest przeprowadzić *serię* eksperymentów, za każdym razem z innym podziałem na L i T. Popularną techniką realizującą to założenie jest *n_{CV}-krotna walidacja skrośna* (lub *krzyżowa*, ang. *n_{CV}-fold cross validation*, por. np. [Weiss & Kulikowski 1991]), która polega na podziale zbioru obiektów na n_{CV} rozłącznych i możliwie równolicznych podzbiorów T_l i przeprowadzeniu n_{CV} eksperymentów, z których każdy składa się z uczenia klasyfikatora na zbiorze uczącym utworzonym przez połączenie $n_{CV} - 1$ podzbiorów i testowaniu na pozostałym jednym podzbiorze. Wypadkową trafność klasyfikowania oblicza się na podstawie liczby poprawnie zaklasyfikowanych obiektów w poszczególnych eksperymentach, czyli

$$\eta_{CV}(h, X) = \frac{1}{|X|} \sum_{l=1}^{n_{CV}} |T_l| \, \eta(h_l(L_l), T_l), \; L_l = \bigcup_{\substack{i=1 \\ i \neq l}}^{n_{CV}} T_i \tag{4.3}$$

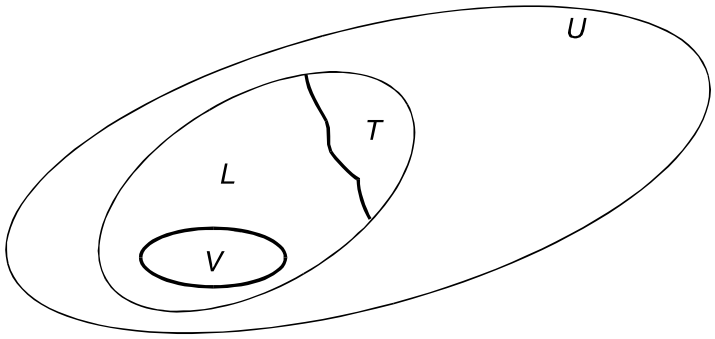
gdzie L_l i T_l są odpowiednio zbiorem uczącym i testującym w l-tej iteracji walidacji skrośnej. Wielkość 4.3 uważa się powszechnie za dobry estymator zdolności predykcyjnej klasyfikatora i stosuje jako wyrażenie maksymalizowane 4.2. Stosowanie walidacji skrośnej dla $n_{CV} = 10$ stało się pewnym standardem w literaturze przedmiotu.

Wyodrębnienie zbioru testującego ma na celu oszacowanie zdolności predykcyjnej klasyfikatora *po* zakończeniu uczenia. Niekiedy istnieje jednak potrzeba oceny trafności klasyfikownia *w trakcie uczenia*. W takim przypadku ze zbioru uczącego L należy wydzielić pewien podzbiór $V \subset L$, $V \cap L = \emptyset$, zwany *zbiorem weryfikującym* (ang. *verification set*). Przez analogię do walidacji skrośnej można rozważać także *weryfikację skrośną*, rozumianą jako wykonanie wielokrotnego eksperymentu uczenia i testowania w ramach zbioru uczącego. Wzajemne relacje poszczególnych podzbiorów przykładów prezentuje Rys. 4.1.

Inne kryteria oceny klasyfikatorów

Zdolność predykcyjna jest najważniejszym, ale nie jedynym kryterium oceny klasyfikatora. Do innych istotnych czynników należą:

- *czytelność* stosowanej reprezentacji wiedzy (implikowana w znacznym stopniu przez przyjęty język reprezentacji hipotez),

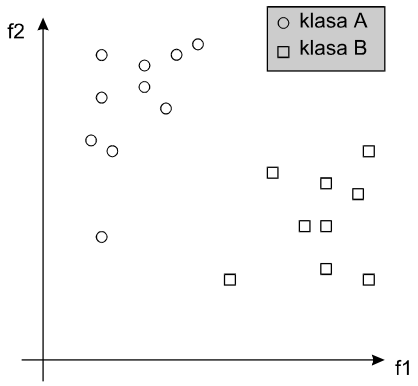


Rysunek 4.1: Relacje pomiędzy podzbiorami przykładów

- *efektywność* procesu
  - *uczenia* - wynikająca ze złożoności obliczeniowej i pamięciowej algorytmu uczenia,
  - *klasyfikacji* - wynikająca ze złożoności obliczeniowej i pamięciowej procedury klasyfikującej,
- zdolność do *adaptacji* (dla problemów o charakterystyce zmieniającej się w czasie).

Hipoteza generowana przez induktor stanowi pewien *opis* zbioru uczącego, który może być *dyskryminujący (odróżniający)* lub *charakterystyczny (charakteryzujący)* (por. np. [Bolc & Zaremba 1992], rozdział 3). Głównym zadaniem opisu dyskryminującego jest możliwie precyzyjne odseparowanie od siebie przykładów reprezentujących poszczególne klasy decyzyjne, podczas gdy opis charakterystyczny precyzyjnie opisuje analizowane pojęcie. Inna linia podziału pomiędzy klasyfikatorami dotyczy *rozmiaru* generowanego opisu (hipotezy). W praktyce często rozważa się minimalne opisy dyskryminujące, pozwalające na szybkie klasyfikowanie przykładów, oraz maksymalne opisy charakteryzyjące, dające pełny opis badanego zjawiska.

Praca niniejsza koncentruje się w znacznej części na opisie dyskryminującym i kryterium zdolności predykcyjnej, jako że dobra ocena na tym kryterium jest warunkiem koniecznym zaakceptowania klasyfikatora w zdecydowanej większości zastosowań praktycznych. Kwestie powiązane z pozostałymi kryteriami zostaną poruszone m.in. w opisie proponowanego podejścia (rozdział 5).



Rysunek 4.2: Przykładowy problem uczenia maszynowego (objaśnienia w tekście)

4.3 Ukierunkowanie indukcyjne (UI)

W poprzednim podrozdziale pokazano, że ograniczona liczność zbioru uczącego w stosunku do populacji wszystkich przykładów implikuje indukcyjny ("ekstrapolacyjny") charakter zadania nadzorowanego uczenia się z przykładów. Z przykładów dostępnych w zbiorze uczącym system musi wyindukować wiedzę trafnie opisującą instancje spoza tego zbioru.

Jednym z podstawowych problemów nadzorowanego uczenia maszynowego jest fakt, że dla danego zbioru uczącego L i zastosowanego języka reprezentacji istnieje z reguły wiele hipotez h stanowiących jego opis dyskryminujący, tj. maksymalizujących z (wyrażenie 4.2). Oznaczmy zbiór takich hipotez przez H_{acc}.

Przykład 1 *Powyższą prawidłowość ilustruje Rys. 4.2 dla prostego zbioru uczącego zawierającego przykłady opisane dwoma atrybutami reprezentujące dwie klasy decyzyjne. Np. dla hipotez wyrażanych w postaci sumy ważonej atrybutów warunkowych (język opisu charakterystyczny dla sztucznych sieci neuronowych) istnieje nieskończenie wiele prostych separujących przykłady z klasy C^+ od przykładów z klasy C^-.*

Z teoretycznego punktu widzenia wielkość zbioru H_{acc} można ograniczyć zwiększając liczność zbioru uczącego. Liczba dostępnych obserwacji jest jednak prawie zawsze ograniczona. Co więcej, nie ma gwarancji, że powiększony zbiór przykładów będzie miał taki rozkład, który ograniczy H_{acc}. Ponadto rozwiązanie takie jest wykonalne jedynie dla problemów o małej liczbie cech, ponieważ liczba przykładów potrzebnych do równomiernego wypełnienia m-wymiarowej przestrzeni cech z ustaloną "gęstością" rośnie wykładniczo wraz ze wzrostem m. Tę niekorzystną własność

przestrzeni przykładów w uczeniu indukcyjnym określa się często mianem *klątwy wymiarowości* (ang. *curse of dimensionality*, [Bellman 1961]).

Przy braku dodatkowych założeń można zatem wygenerować wiele (często nieskończenie wiele) hipotez dyskryminujących (w 100% lub w jakimś założonym stopniu) przykłady należące do zbioru uczącego L. Jednak tylko niektóre z tych hipotez prawidłowo klasyfikują obiekty spoza zbioru uczącego. induktor musi być zatem wyposażony w pewne ograniczenia i/lub kryteria, które pozwolą na wybór jednej hipotezy z H_{acc}.

Pewne ograniczenia wynikają z rozmiaru zbioru H, który jest zazwyczaj bardzo obszerny, a w przypadku niektórych typów algorytmów indukcji nieskończony. Dla większości stosowanych języków reprezentacji złożoność obliczeniowa procesu pełnego przeszukiwania przestrzeni hipotez jest wykładnicza w funkcji liczby atrybutów m. Na przykład już dla prostego przypadku m cech binarnych (dwuwartościowych) i reprezentowania hipotez przez koniunkcje wartości cech liczność przestrzeni hipotez wynosi $|H| = 2^m$. Stąd w praktyce do przeszukiwania H stosuje się różne metaheurystyki i heurystyki, na przykład metodę spadku gradientu w sztucznych sieciach neuronowych (por. np. [Żurada, Barski, *et al.* 1996]), "stromą" optymalizację w generowaniu reguł i drzew decyzyjnych [Quinlan 1979], itp. W konsekwencji w procesie uczenia induktor przegląda jedynie niewielką część zbioru H.

Inne ograniczenia stosowane w algorytmach indukcji wynikają z obserwacji, że rzeczywiste problemy uczenia maszynowego w większości charakteryzują się regularną strukturą. Regularność ta przejawia się przede wszystkim w częstym skupianiu się przykładów należących do poszczególnych klas decyzyjnych w różnych częściach przestrzeni przykładów (por. Rys. 4.2). Innym przejawem tej regularności jest wielomodalność rozkładów cech f_j, gdzie mody reprezentują wartości cechy charakterystyczne dla poszczególnych klas decyzyjnych.

Systemy uczące się korzystają z wiedzy o różnych (nie tylko tych wymienionych wyżej) cechach charakterystycznych problemów uczenia maszynowego, aby zwiększyć prawdopodobieństwo wygenerowania hipotezy, która będzie satysfakcjonująco klasyfikować także przykłady spoza zbioru uczącego. Tę własność systemów uczących się określa się mianem *ukierunkowania indukcyjnego* (UI).

Definicja 2 Ukierunkowanie indukcyjne *(ang.* inductive bias*) to zbiór wszystkich czynników, które wpływają na wybór hipotezy przez system uczący się na danym zbiorze przykładów X. Czynniki te obejmują definicję przestrzeni hipotez H oraz definicję algorytmu jej przeszukiwania [za [Utgoff 1983], str. 5].*

Ukierunkowanie indukcyjne jest z reguły implementowane w systemie uczącym w postaci *ograniczeń* lub *kryteriów* oceny hipotez, które wynikają z wyżej przedstawionych rozważań lub mają charakter "zdroworozsądkowy". Najbardziej elementarnym

ukierunkowaniem indukcyjnym obecnym w niemal wszystkich typach algorytmów indukcji jest *założenie ciągłości* (ang. *continuity assumption*), które mówi, że obiekty bliskie w przestrzeni cech należą zazwyczaj do tej samej klasy decyzyjnej. Inne ważne ukierunkowanie polega na preferowaniu hipotez prostszych (ang. *simplicity bias*), czyli np. mniejszych zbiorów reguł, krótszych reguł, drzew decyzyjnych o mniejszej liczbie węzłów decyzyjnych, etc.

Ukierunkowanie indukcyjne jest zazwyczaj następstwem (za [Whigham 1996]):

1. sposobu oceny hipotez i preferowania wyboru pewnych hipotez nad inne (ang. *selection bias*),
2. reprezentacji wiedzy stosowanej przez induktor (ang. *language bias*),
3. sposobu przeszukiwania przestrzeni hipotez (ang. *search bias*).

Przykład 3 *W przypadku drzew decyzyjnych (np. ID3, [Quinlan 1979]) ograniczenie typu 1) wynika z faktu, iż wiedza reprezentowana jest w postaci drzewa decyzyjnego, ukierunkowanie typu 2) jest następstwem zachłannej strategii generowania drzewa (*best-first*), zaś ukierunkowanie typu 3) manifestuje się w warunkach stopu, które na przykład nie pozwalają rozwijać węzłów drzewa o zbyt małej liczności lub węzłów o niekorzystnym rozkładzie przypadków w poszczególnych klasach decyzyjnych.*

W literaturze (np. [Bensusan 1998]) spotkać można nieformalny podział ukierunkowań na *silne* (*strong*) i *słabe* (*weak*). Ukierunkowania silne narzucają algorytmowi uczącemu się generowanie pewnych hipotez nawet wtedy, gdy liczność zbioru uczącego jest niewielka. Z kolei ukierunkowanie słabe pozwala na to, aby przebieg procesu uczenia był bardziej zależny od przykładów ze zbioru uczącego.

Inny istotny podział ukierunkowań to rozróżnienie ukierunkowań *twardych* (ang. *hard/representation bias*) i *miękkich* (ang. *soft/preference/procedural bias*). Ukierunkowanie twarde uniemożliwia systemowi uczącemu się wygenerowanie pewnych hipotez, natomiast ukierunkowanie miękkie definiuje pewne preferencje lub porządek (np. w postaci rozkładu prawdopodobieństwa) w zbiorze generowanych hipotez, czyli powoduje, że wygenerowanie pewnych hipotez jest bardziej prawdopodobne niż wygenerowanie innych. Ukierunkowania twarde implementowane są przez ograniczenia, a miękkie - przez kryteria oceny hipotez. Przykładem ukierunkowania miękkiego jest preferowanie prostszych (krótszych) hipotez, a twardego - narzucenie ograniczenia na długość reguły decyzyjnej.

Ukierunkowanie indukcyjne wiąże się silnie z problemem przeuczenia. W literaturze statystycznej związek zjawiska przeuczenia z ukierunkowaniem indukcyjnym

znany jest pod terminem *dylematu ukierunkowanie-zmienność (elastyczność)* (ang. *bias-variance dilemma*, [Geman & Bienenstock 1992]). Przez pojęcie to rozumie się przetarg, jaki zachodzi pomiędzy ukierunkowaniem indukcyjnym a elastycznością (wariancją) systemu uczącego się. System o słabym ukierunkowaniu indukcyjnym ma dużą elastyczność, tj. jest w stanie dokładnie opisać przykłady ze zbioru uczącego. Jednak w związku z tym może on mieć tendencje do przeuczenia. Chcąc temu zapobiec, możemy wbudować w system uczący się silniejsze ukierunkowanie, co z kolei pociąga za sobą zmiejszenie elastyczności algorytmu indukcji, który w konsekwencji może nie być w stanie wygenerować odpowiedniej hipotezy.

4.3.1 W poszukiwaniu optymalnego UI

Wczesne etapy badań w uczeniu maszynowym naznaczone były znacznym optymizmem, wyrażającym się w poszukiwaniu uniwersalnych algorytmów indukcji, tj. takich, które wykazują się dobrą skutecznością predykcyjną w wielu zadaniach. Obecnie uznaje się, że nie istnieje uniwersalne (optymalne) ukierunkowanie indukcyjne. Milowym krokiem na tej drodze były prace Wolperta [Wolpert 1996], [Wolpert 1996], który wykazał m.in., iż każdy induktor (lub, bardziej ogólnie, każda heurystyka) charakteryzuje się taką samą skutecznością w przypadku średnim, tj. na populacji wszystkich możliwych instancji problemu (tzw. twierdzenie *'no free lunch'* [Wolpert & Macready 1995]).

W świetle tych spostrzeżeń można stwierdzić, iż powszechnie używane metody uczenia maszynowego uzyskiwanie dobrej skuteczności w przypadku średnim zawdzięczają głównie temu, że rzeczywiste problemy są w pewnym sensie do siebie podobne. Poszczególne metody eksploatują to podobieństwo na różne sposoby, zależnie od ukierunkowania indukcyjnego, jakim się posługują. W związku z tym w ostatnich latach intensywnie rozwija się podejście *uczenia wielostrategicznego* (ang. *multistrategy learning*, por. [Michalski & Tecuci 1994]), wykorzystując:

- integrację algorytmów indukcji różnych typów (np. drzew decyzyjnych, specjalizowanego algorytmu konstruktywnej indukcji i sieci neuronowej [Hunter 1996]),
- zmianę ukierunkowania w trakcie uczenia (ang. *dynamic bias*)[5],

[5] Podział systemów uczących się na systemy ze stałym i zmiennym ukierunkowaniem jest raczej orientacyjny i nieformalny. Trudno jest jednoznacznie wyznaczyć granicę między nimi. Na przykład, czy algorytm uczenia sieci neuronowej, w którym prędkość uczenia jest zmniejszana (np. hiperbolicznie) w kolejnych epokach, należy jeszcze do uczenia ze stałym ukierunkowaniem, czy też już ze zmiennym?

- pozyskiwanie ukierunkowania z doświadczenia (ang. *learned bias*) z różnymi zastosowaniami (zbiorami przykładów X), np. [Bensusan 1998].

Konstruktywna indukcja cech opisywana w kolejnym rozdziale jest przykładem metodyki, w której dzięki modyfikacjom przestrzeni cech dochodzi do osłabienia ukierunkowania indukcyjnego. W konsekwencji induktor zyskuje możliwość wygenerowania hipotez, które są dla niego "nieosiągalne", gdy posługuje się oryginalną przestrzenią reprezentacji.

4.4 Konstruktywna indukcja cech (KI)

Podrozdział ten opisuje zagadnienie konstruktywnej indukcji cech (KI). W znacznym stopniu wykorzystywane są tu także pojęcia charakterystyczne dla selekcji cech, ponieważ jest to zagadnienie lepiej rozpoznane i częściej spotykane w literaturze uczenia maszynowego.

4.4.1 Geneza KI

Konwencjonalne metody uczenia maszynowego czynią z reguły założenie o niezmienności reprezentacji przykładów, w której odbywa się poszukiwanie hipotezy. Ujmując to bardziej precyzyjnie, zbiór cech oryginalnych F_0 opisujących przykłady nie jest modyfikowany w trakcie uczenia. W literaturze anglojęzycznej własność ta charakteryzowana jest niekiedy określeniem *fixed feature repertoire* [Schyns, Goldstone, *et al.* 1997].

W prostych zastosowaniach uczenia maszynowego oryginalny zestaw cech może być wystarczający, tj. zapewniający satysfakcjonującą zdolność dyskryminacyjną (zakładamy skupienie się na opisie dyskryminującym). Jednak w trudnych problemach może okazać się, że przy użyciu oryginalnego zestawu cech i wybranego języka reprezentacji wiedzy nie da się wygenerować hipotezy o zadowalającej trafności klasyfikowania ($h \in H_{acc}$), lub że wygenerowana hipoteza będzie bardzo rozbudowana, co pociąga za sobą z reguły słabe zdolności uogólniania.

Dobrym, choć co prawda sztucznym przykładem jest tu znany zwłaszcza w dziedzinie sztucznych sieci neuronowych problem dwóch spiral, gdzie wartości dwóch cech wygenerowane są według schematu $(c\cos(n\alpha), c\sin(n\alpha))$ dla pierwszej klasy decyzyjnej i $(c\cos(n\alpha+\pi), c\sin(n\alpha+\pi))$ dla drugiej klasy decyzyjnej (gdzie c i α to pewne stałe, a n to numer przykładu). Większości algorytmów indukcji trudno jest wygenerować dobrą hipotezę dla tego problemu przy oryginalnej reprezentacji przykładów (współrzędne kartezjańskie). Natomiast po zmianie przestrzeni reprezentacji na biegunowy układ współrzędnych staje się on niemal trywialny.

Odporność na tego typu sytuacje zależy w znacznej mierze od *typu* algorytmu indukcji, a dokładniej od tego, w jakim stopniu i w jaki sposób potrafi on wykorzystywać synergię cech. Większość konwencjonalnych algorytmów indukcji tworzy oczywiście hipotezy w *wielowymiarowej* przestrzeni opisu. Jednak powiązania cech istotne dla danego problemu klasyfikacji mają niekiedy na tyle skomplikowany charakter, że induktor nie jest w stanie "odkryć" hipotezy wyjaśniającej dane. Właśnie to ograniczenie systemów uczenia maszynowego stało się główną przesłanką dla metod konstruktywnej indukcji cech. Konstruktywna indukcja cech jest w gruncie rzeczy próbą automatyzacji procesu doboru/wyboru reprezentacji (ang. *choice of representation space*), który jest zazwyczaj udziałem projektanta systemu uczącego się.

Teza o konieczności konstrukcji cech w trakcie uczenia znajduje także potwierdzenie w pracach z pogranicza sztucznej inteligencji i psychologii poznawczej (por. np. [Tomaszewski 1992], [Wiśniewski & Medin 1994]). W pracy [Schyns, Goldstone, *et al.* 1997] pokazuje się na przykład, jak w trakcie rozwiązywanania problemu człowiek wykształca nowe cechy i jak efektywność (mierzona np. czasochłonnością) procesu rozpoznawania zależy od możliwości ich generowania.

Należy podkreślić, iż konstruktywna indukcja cech to jedynie jedna z wielu metod modyfikowania przestrzeni reprezentacji obecnych w uczeniu maszynowym. Także wiele algorytmów uczenia maszynowego uznawanych za konwencjonalne dokonuje modyfikacji przestrzeni reprezentacji w niejawny sposób (patrz punkt 4.4.10).

4.4.2 Konstrukcja cech

Konstrukcja cech (ang. *feature construction*) to proces mający na celu uzyskanie nowych cech z oryginalnego opisu [Matheus & Rendell 1989]. Konstruowanie nowej cechy odbywa się poprzez zastosowanie pewnej funkcji (operatora) do wybranych cech z oryginalnego opisu F_0. Zgodnie z tą definicją, konstrukcja cech nie wprowadza nowej informacji (w sensie teorii informacji), ponieważ nowo utworzony atrybut jest zależny funkcyjnie od cech oryginalnych.

W pracy [Matheus & Rendell 1989] zaproponowano podział konstrukcji cech na dwa zagadnienia w zależności od kontekstu:

- konstruktywna *kompilacja* cech (ang. *constructive compilation*),
- konstruktywna *indukcja* cech (ang. *constructive induction*).

Konstruktywna kompilacja cech ogranicza się jedynie do tworzenia nowych cech, których zadaniem jest lepsze "wyjaśnianie" danego zbioru przykładów (np. dyskryminowanie pomiędzy klasami decyzyjnymi). Aspekt uogólniania jest więc w tym

podejściu pominięty (nie jest przynajmniej uwzględniany w sposób jawny). Podobne zagadnienie spotyka się w gałęzi sztucznej inteligencji zwanej odkrywaniem praw (ang. *computer discovery*, zob. np. [Michalski, Carbonell, *et al.* 1983], [Zembowicz & Żytkow 1991], [Zembowicz & Żytkow 1992]).

Natomiast niezbywalnym przymiotem konstruktywnej *indukcji* cech jest uwzględnienie indukcyjnego charakteru tworzonej reprezentacji [Michalski, Carbonell, *et al.* 1983]. Nowe cechy mają, poza wyjaśnianiem przykładów ze zbioru uczącego, polepszyć *zdolności generalizacyjne* systemu uczącego się przez wykrywanie w analizowanych problemach hipotez "wyższego rzędu". To oczekiwanie jest w pewnym stopniu uzasadnione, ponieważ nowe cechy prowadzą zazwyczaj do *uproszczenia hipotezy* (np. skrócenia jej długości), a proste hipotezy są zazwyczaj bardziej ogólne (porównaj uwagi nt. ukierunkowania na hipotezy proste w punkcie 4.3). Nacisk kładziony na poszczególne cele KI w dużej mierze zależy od specyfiki rozważanego zastosowania. Na przykład w zastosowaniach, w których wyjaśnianie podejmowanych decyzji jest szczególnie ważne, znaczenie drugiego z wyżej wymienionych celów będzie większe niż w aplikacjach nie stawiających tego wymagania.

4.4.3 Fazy KI

Konstruktywna indukcja przeprowadzana jest zazwyczaj iteracyjnie, prowadząc do kolejnych zmian przestrzeni reprezentacji. Niech F_k oznacza zbiór cech (przestrzeń reprezentacji) wygenerowany w k-tej iteracji KI. Wówczas KI stanowi pewien ciąg reprezentacji $F_0, F_1, ..., F_k, ..., F_{k^+} = F^+$, gdzie F_0 jest reprezentacją oryginalną. W ramach pojedynczego etapu KI, polegającego na transformacji bieżącej przestrzeni reprezentacji F_k do kolejnej przestrzeni F_{k+1}, wyróżnia się w ogólności następujące fazy (za [Matheus & Rendell 1989]):

1. faza detekcji (ang. *detection*),
2. faza selekcji (ang. *selection*),
3. faza uogólnienia (ang. *generalisation*),
4. faza oceny (ang. *evaluation*).

Ad. 1) W *fazie detekcji* podejmuje się decyzję *czy* należy modyfikować bieżącą przestrzeń reprezentacji. Decyzja ta podejmowana jest zazwyczaj przez system (choć niektóre metody KI uwzględniają na tym etapie decyzje ekperta) w oparciu o pewne kryteria oceny bieżącego rozwiązania F_k i (w podejściu HCI, por. punkt 4.4.7) analizę hipotezy wygenerowanej dla tej reprezentacji.

Ad. 2) W *fazie selekcji* wybrany zostaje *operator konstruktywny* o_i (ang. *constructive operator*) i *operand konstruktywny* $F' \subseteq F_k$ (ang. *constructive operand*). Operator o_i wybierany jest zgodnie z obraną strategią ze zdefiniowanego przez eksperta zbioru operatorów O. Operand F' jest zazwyczaj pojedynczą cechą lub parą cech, gdzie wielkość $\left|F'\right|$ nazywać będziemy *rozmiarem operandu*. Poprzez zastosowanie o_i do F' powstaje nowa cecha f_j, co zapisujemy $f_j \leftarrow o_i(F')$. Nowo utworzona cecha jest dołączana do bieżącego zbioru cech, co prowadzi do powstania nowej przestrzeni reprezentacji $F_{k+1} = F_k \cup \{f_j\}$. W procesie KI faza selekcji jest krytycznym punktem, ponieważ w niej rozstrzyga się na podstawie których cech i w jaki sposób ma być utworzona nowa cecha. Wybór odpowiedniego operandu i operatora konstruktywnego jest z reguły bardzo trudny i wymaga dodatkowej wiedzy (por. punkt 4.4.7).

Ad. 3) *Faza uogólniania* polega na uczeniu wybranego algorytmu indukcji na zmodyfikowanym opisie F_{k+1} i jest zasadniczo realizowana jedynie w podejściu konstruktywnej indukcji cech sterowanej hipotezą (*HCI*, por. punkt 4.4.7), gdzie do podjęcia decyzji o utworzeniu nowej cechy niezbędne jest wygenerowanie nowej hipotezy. Ponieważ większość metod uczenia maszynowego jest bardzo wrażliwa na zmiany reprezentacji, etap ten wymaga zazwyczaj przeprowadzenia nowego procesu uczenia. Niedogodność ta stanowi główną przyczynę znacznej złożoności obliczeniowej HCI.

Ad. 4) W *fazie oceny* otrzymane rozwiązanie F_{k+1} podlega ocenie ze względu na wybrane własności. Do kryteriów najczęściej stosowanych w tej fazie KI należą estymata zdolności predykcyjnej i/lub miara długości hipotezy. Ponieważ sposób oceniania reprezentacji ma kluczowe znaczenie dla KI, zagadnieniu temu poświęcony jest cały kolejny podrozdział.

4.4.4 Funkcja oceniająca

Niech E oznacza funkcję stosowaną w procesie KI do oceny podzbioru cech. Bez utraty ogólności przyjmijmy, że jest ona maksymalizowana i nazwijmy ją *funkcją oceniającą*. Wartość $E(F)$ odzwierciedla stopień spełnienia przez zbiór cech F jednego z lub obu kryteriów wymienionych w poprzednim podrozdziale, tj. zdolności predykcyjnej i rozmiaru hipotezy. Zazwyczaj zakłada się że $E(\varnothing) = 0$; w niektórych podejściach przyjmuje się także $E(F^*) = 1$, gdzie F^* jest reprezentacją optymalną.

W ogólności wszystkie funkcje oceniające można podzielić na dwie kategorie:

- *monotoniczne*, tj. spełniające warunek słabej monotoniczności[6]:

$$\forall\ F_i, F_j : F_i \subset F_j \Rightarrow E(F_i) \leq E(F_j) \tag{4.4}$$

- *niemonotoniczne*, tj. takie, które nie spełniają warunku 4.4.

Podział ten ma fundamentalne znaczenie dla strategii przeszukiwania przestrzeni stanów. Relacja słabej monotoniczności określa bowiem *częściowy porządek* w przestrzeni rozwiązań/stanów (por. punkt 4.4.5), co może mieć korzystne implikacje dla algorytmu przeglądania tej przestrzeni. Mamy wówczas pewność, że usuwanie cech z F w najlepszym wypadku zachowa jej wartość (a z pewnością nie zwiększy). W konsekwencji algorytm może a priori zrezygnować z przeszukiwania pewnych podzbiorów stanów, ponieważ pewne jest, że nie znajdzie w nich rozwiązań lepszych od tych już odwiedzonych. Daje to możliwość znalezienia rozwiązania (rozwiązań) dokładnego w wielomianowym czasie, np. przy użyciu znanego z badań operacyjnych algorytmu podziału i ograniczeń (ang. *branch and bound*), por. np. [Narendra & Fukunaga 1997], [Cios, Pedrycz, *et al.* 1998], [Błażewicz, Cellary, *et al.* 1983] s. 470.

Jednak w praktyce monotoniczne funkcje oceniające dają zazwyczaj gorsze rezultaty niż funkcje niemonotoniczne (por. np. [Dash & Liu 1997]). Wynika to z faktu, iż większość z nich odzwierciedla jedynie stopień dyskryminowania przez zbiór cech przykładów ze zbioru uczącego, ignorując kluczową dla uczenia maszynowego konieczność uogólniania posiadanej wiedzy. Natomiast zdolność predykcyjna klasyfikatora pozostaje często w sprzeczności z warunkiem monotoniczności (4.4), tj. może być lepsza dla pewnego podzbioru cech F_i niż dla $F_j \supset F_i$.

Opisywane w literaturze metody KI i selekcji cech można podzielić na (por. [Langley 1994], [Dash & Liu 1997], [Cios, Pedrycz, *et al.* 1998] rozdział 9)

1. metody typu *filter* (ang. *filter, open-loop, preset bias, front-end*), dające wyniki niezależne od algorytmów indukcji. W metodach tych funkcja E przyjmuje zazwyczaj wartości obliczone na podstawie
   (a) odległości pomiędzy przykładami w F (ang. *distance*),
   (b) separowalności przykładów w F (ang. *[interclass] separability, discrimination*),
   (c) zawartości informacyjnej zbioru F (ang. *[mutual] information*),
   (d) zależności pomiędzy cechami w F (ang. *dependence*),

[6] Monotoniczność jest jednym z tzw. warunków Choquet; ich spełnienie implikuje, że E staje się tzw. miarą rozmytą (ang. *fuzzy measure*)

(e) zgodności wartości cech z F z podziałem na klasy decyzyjne (*spójność*, ang. *consistency*),

2. metody typu *wrapper* (ang. *closed-loop, performance bias, classifier feedback*), których wynik zależy od algorytmu indukcji zastosowanego do obliczenia wartości funkcji E.

Miary typu 1b i 1e są monotoniczne z definicji. Pozostałe grupy funkcji wymienionych w punkcie 1 reprezentowane są przez funkcje zarówno monotoniczne, jak i niemonotoniczne. Natomiast metody typu *wraper* wykorzystują zazwyczaj trafność klasyfikowania uzyskaną z użyciem wybranego algorytmu indukcji, która jest ze swej natury niemonotoniczna.

W szczególności, z konkretnych miar wykorzystywanych w podejściach typu *filter* wymienić należy:

- zysk na informacji (ang. *information gain*) oparty na entropii, stosowany często w indukcji drzew decyzyjnych (typ 1c) [Quinlan 1992],
- minimalną długość opisu (ang. *minimum description length*, MDL, [Rissanen 1983], typ 1c, ale pośrednio bierze także pod uwagę rozmiar zbioru F, stąd niemonotoniczna) ,
- jakość (przybliżenia) klasyfikacji - wielkość opisująca tzw. system informacyjny w teorii zbiorów przybliżonych (ang. *rough sets theory* [Pawlak 1982], [Pawlak 1991], [Słowiński 1992], [Słowiński 1995], [Pawlak & Słowiński 1994], typ 1e).

Podejście *wrapper* [John, Kohavi, *et al.* 1994], [Kohavi & John 1997] zdobyło sobie ostatnio dużą popularność, do czego przyczynił się zapewne głównie wzrost mocy obliczeniowej komputerów, dzięki czemu jego wykorzystanie w praktyce stało się bardziej realne. Jego oryginalność polega na użyciu trafności klasyfikowania algorytmu uczenia jako funkcji oceniającej E. Zbiór uczący jest w tym celu dzielony na kilka podzbiorów i przeprowadzane jest na nim wielokrotne uczenie i testowanie (najczęściej metodą walidacji skrośnej, por. rozdział 4). Za wartość funkcji $E(F)$ przyjmuje się trafność klasyfikowania uzyskaną w tym eksperymencie przy użyciu zbioru cech F (wyrażenie 4.3).

Podstawową zaletą takiego estymowania przydatności zbioru cech na podstawie "wewnętrznego", wielokrotnego eksperymentu uczenia i testowania jest uwzględnienie ukierunkowania indukcyjnego stosowanego algorytmu (choć może to być wada, jeśli poszukujemy zbioru cech niezależnego od wybranego algorytmu indukcji). Do

wad należy wysoka złożoność obliczeniowa wynikająca z konieczności wielokrotnego przeprowadzenia uczenia dla każdego ocenianego zbioru cech F. Jednak ten zwiększony koszt się opłaca - jak pokazują eksperymenty (por. np. [Dash & Liu 1997]) *wrapper* wydaje się być obecnie najlepszą znaną funkcją oceniającą dla selekcji cech.

4.4.5 KI jako przeszukiwanie przestrzeni stanów

KI dogodnie jest sformułować jako problem przeszukiwania przestrzeni stanów, gdzie każdy stan odpowiada pewnemu zbiorowi cech F_i, zaś zastosowania operatorów konstruktywnych o_i przeprowadzają system uczący się ze stanu w stan. Proces przeszukiwania ukierunkowany jest wartościami funkcji E dla poszczególnych podzbiorów cech. Niech $\mathcal{F}(F_0, O)$ (lub krótko $\mathcal{F}$)[7] oznacza zbiór wszystkich stanów (reprezentacji, zbiorów cech) możliwych do uzyskania z oryginalnej reprezentacji F_0 w procesie KI wykorzystującym zbiór operatorów O.

W takim ujęciu system uczenia maszynowego wyposażony w KI realizuje dwa przeplatające się (najczęściej zagnieżdżone jeden w drugim) procesy przeszukiwania przestrzeni:

- przeszukiwanie *przestrzeni reprezentacji* (stan odpowiada pewnemu zbiorowi cech F_i),
- przeszukiwanie *przestrzeni hipotez* H w ramach danej reprezentacji F_i.

Strategie przeszukiwania przestrzeni stanów $\mathcal{F}$ dla potrzeb KI można podzielić na (za [Dash & Liu 1997]):

- dokładne (pełne),
- heurystyczne (w tym także losowe).

Zaletą strategii pełnego przeszukiwania jest pewność otrzymania rozwiązania dokładnego F^*, tj. podzbioru cech optymalizującego przyjęte kryterium E. Oczywistą wadą jest wykładnicza złożoność obliczeniowa, niezależnie od złożoności obliczeń wymaganych przez funkcję oceniającą E. Złożoność ta wynosi w ogólności $O(|\mathcal{F}|)$, a na przykład dla prostszego od KI problemu selekcji cech $O\left(2^{|F_0|}\right)$.

Strategie heurystyczne nie przeglądają wszystkich możliwych podzbiorów cech i w konsekwencji nie dają gwarancji znalezienia rozwiązania optymalnego F^*. W

[7] Zbiór $\mathcal{F}$ dogodnie byłoby nazywać przestrzenią reprezentacji (w sensie: wielu reprezentacji $F' \in \mathcal{F}$), jednak jednakowe brzmienie dopełniacza liczby pojedynczej i mnogiej rzeczownika "reprezentacja" mogło by tu prowadzić do nieporozumień (por. definicja przestrzeni reprezentacji F, str. 29).

praktyce ich wynikiem jest pewne rozwiązanie *suboptymalne* F^+ ($E(F^+) \leq E(F^*)$), stanowiące zazwyczaj *minimum lokalne* w przestrzeni $\mathcal{F}$. Ponieważ stosowane są tu zazwyczaj strategie zachłanne, liczba przeglądanych podzbiorów zależy wielomianowo od $|\mathcal{F}|$. Zatem, o ile obliczanie wartości E nie ma złożoności wykładniczej, strategia heurystyczna ma złożoność wielomianową.

4.4.6 KI a selekcja cech

Selekcja cech (ang. *feature/attribute selection*, por. przeglądy [Dash & Liu 1997], [Langley 1994], również [Domingos 1997]) jest szczególnym przypadkiem KI. Celem selekcji cech jest wyeliminowanie z oryginalnego opisu cech nadmiarowych (ang. *redundant*) i nieistotnych (ang. *irrelevant*). Cechą nadmiarową może być na przykład atrybut zależny funkcyjnie od innej cechy (np. przez wymnożenie przez stałą), zaś przykładem cechy nieistotnej - zmienna losowa której rozkład nie zależy od zmiennej decyzyjnej. O związku KI z zagadnieniem selekcji cech warto wspomnieć m.in. z tej racji, że selekcja cech, jako w ogólności zagadnienie prostsze, jest częściej stosowana w praktyce i opisywana w literaturze.

Pokrewność zagadnień KI i selekcji cech można zilustrować na co najmniej dwa sposoby. Selekcja cech może być potraktowana jako szczególny przypadek indukcji konstruktywnej, w którym operatory nie definiują nowych cech, a jedynie wybierają poszczególne cechy do opisu.

Z drugiej strony KI da się do pewnego stopnia zrealizować w ramach selekcji cech. Wystarczy w tym celu przy użyciu operatorów konstruktywnych wygenerować na operandach z F_0 zbiór F wszystkich cech (oczywiście "wszystkich" przy pewnych ograniczeniach). Na tak poszerzonej reprezentacji należy następnie przeprowadzić selekcję cech. To spostrzeżenie ma jednak jedynie wartość teoretyczną, w praktyce bowiem nawet dla małych zbiorów cech i niewielkiej liczby operatorów KI liczba wygenerowanych cech osiąga zawrotne wielkości. Już dla jednego dwuargumentowego (tj. akceptującego operandy o liczności 2) operatora zastosowanego do oryginalnego opisu składającego się z N cech liczba ta wynosi $\binom{N}{2}$, czyli np. 45 dla $N = 10$. Aby ominąć pojawiający się tu problem wykładniczej złożności obliczeniowej, metody KI stosują wyrafinowane sposoby integrowania mechanizmu zmiany reprezentacji z procesem uczenia, co prowadzi do zawężenia zbioru generowanych cech.

Formalnie problem selekcji cech można ująć następująco: dla danego zbioru cech F_0 znaleźć podzbiór cech $F^* \subseteq F_0$ optymalizujący (maksymalizujący) pewną miarę jakości E:

$$F^* = \arg \max_{F' \subseteq F_0} E(F')$$

Przy braku dodatkowych założeń co do charakteru funkcji E znalezienie dokład-

nego rozwiązania problemu selekcji cech wymaga w ogólności przejrzenia wszystkich $2^{|F_0|}$ podzbiorów oryginalnego zbioru cech F_0, jest zatem problem NP-zupełnym. Stąd najczęściej rezygnuje się z poszukiwania globalnego optimum F^* i w celu zmniejszenia złożoności obliczeniowej powszechnie stosuje się podejścia heurystyczne. Znaczna część podejść przybliżonych używa jednej z dwóch strategii przeszukiwania :

- *Wybieranie wstępujące* (ang. *forward selection*). Selekcja rozpoczyna się od pustego zbioru cech i odbywa się przez dodawanie cech do opisu.
- *Eliminacja zstępująca* (ang. *backward elimination*). W tym przypadku proces rozpoczyna się od pełnego zbioru cech i postępuje przez usuwanie kolejnych cech z opisu.

Poza tym w literaturze spotkać można bardziej wyrafinowane metody przeglądania przestrzeni stanów, do których zaliczyć można podejścia hybrydowe (np. [Moore & Lee 1994]), probabilistyczne (np. [Liu & Setiono 1997]), czy wykorzystujące algorytmy ewolucyjne [Vafaie & Imam 1994]. Obszerny przegląd technik selekcji cech znaleźć można m.in. w pracy [Dash & Liu 1997]. Podejście proponowane w niniejszej pracy w jednej ze swych odmian implementuje technikę wybierania wstępującego poprzez algorytm stromego przeszukiwania lokalnego SLS (patrz rozdział 5).

4.4.7 Sterowanie procesem KI

W pracy [Wnek & Michalski 1994] zaproponowano podział metod KI na następujące cztery grupy ze względu na sposób sterowania procesem KI, w tym zwłaszcza sposób wyboru operandu i operatora konstruktywnego w fazie selekcji:

- KI sterowana danymi (ang. *data-driven CI, DCI*),
- KI sterowana wiedzą (ang. *knowledge-driven CI, KCI*),
- KI sterowana hipotezą (ang. *hypothesis-driven CI, HCI*),
- wielostrategiczna KI (ang. *multistrategy CI, MCI*).

W DCI decyzja o modyfikacji przestrzeni cech i o sposobie zdefiniowania nowej cechy zapada na podstawie analizy danych, tj. zbioru uczącego L. Szczególnie przydatne może być tu wykrywanie zależności funkcyjnych i współzmienności poszczególych cech (mierzonych na przykład za pomocą miar statystycznych takich jak współczynnik korelacji czy χ^2).

W przypadku KCI wiedza o tym kiedy i jak konstruować nowe cechy jest (przynajmniej do pewnego stopnia) zawarta w bazie wiedzy systemu, utworzonej a priori przez projektanta systemu. Na wiedzę tę zazwyczaj składają się fakty dotyczące dziedziny zastosowania (ang. *domain knowledge*) oraz fakty wynikające z szeroko rozumianego "zdrowego rozsądku" (ang. *common sense knowledge*).

HCI stanowi chyba najbardziej wyrafinowaną odmianę KI. Główną przesłanką dla tej grupy podejść jest dość oczywiste spostrzeżenie, że zmiany reprezentacji odbijają się na kształcie wygenerowanej hipotezy. Zatem, zamiast analizować przykłady w przestrzeni cech (co może mieć znaczącą złożoność obliczeniową), w HCI podejmuje się próbę wnioskowania o potrzebie i sposobie modyfikacji przestrzeni cech na podstawie samej hipotezy. Takie podejście da się oczywiście zastosować głównie tam, gdzie reprezentacja hipotezy jest jawna i czytelna (np. dla reguł i drzew decyzyjnych) i gdzie w konsekwencji możliwe jest zdefiniowanie pewnych warunków wyzwalających odpowiednie procesy KI.

Z kolei podejścia typu MCI mają charakter hybrydowy i czerpią w różnym stopniu z powyższych trzech strategii KI. Próby podejmowane w tym nurcie KI wyrastają ze spostrzeżenia, że nie istnieje uniwersalne ukierunkowanie indukcyjne i induktor najlepszy we wszystkich możliwych zastosowaniach (por. podrozdział 4.3). W związku z tym w MCI łączy się różne podejścia do KI z nadzieją, że dojdzie do synergii ich mocnych stron.

4.4.8 Operatory KI

Operator konstruktywny o_i określa, jak skonstruować nową cechę na podstawie operandu konstruktywnego F'. Zależnie od rozmiaru operandu (tj. liczby składających się na niego cech $|F'|$), typu cech wchodzących w jego skład oraz typu tworzonej cechy istnieje bardzo wiele (często nieskończenie wiele) różnych operatorów. Rozważanie dowolnych operatorów nie jest w związku z tym możliwe ze względów obliczeniowych, a ponadto nie ma też sensu z praktycznego punktu widzenia, jako że pożądane jest stosowanie operatorów posiadających "rozsądną" interpretację. Zbiór O operatorów zastosowanych w konkretnym problemie KI jest zatem ograniczony i implementuje pewną dodatkową *wiedzę* dla systemu uczącego się. Wiedza ta ma zazwyczaj charakter "zdroworozsądkowy" (ang. *common sense knowledge*) lub "dziedzinowy" (ang. *domain knowledge, background knowledge*). Istotna część procesu selekcji (por. punkt 4.4.3) ma zazwyczaj miejsce jeszcze przed procesem KI i polega na zaimplementowaniu w systemie KI tylko wybranych operatorów[8].

Zbiór O operatorów zastosowanych w konkretnym problemie KI może być dobrany w sposób

[8] Można zatem stwierdzić, iż każda metoda KI jest po części sterowana wiedzą (KCI)

- niezależny od dziedziny (ang. *domain-independent*) lub
- zależny od dziedziny (ang. *domain-dependent*).

Zaletą pierwszego sposobu doboru operatorów jest jego uniwersalność i łatwość konstrukcji (np. proste operatory logiczne i arytmetyczne). Natomiast dobór operatorów dla konkretnego zadania jest bardziej czasochłonny, ale tak pozyskane operatory dają większe prawdopodobieństwo wygenerowania użytecznych cech.

Ze względu na sposób modyfikowania reprezentacji operatory KI dzieli się przede wszystkim na

- operatory rozszerzające przestrzeń reprezentacji (ang. *extension operators*),
- operatory zawężające przestrzeń reprezentacji (ang. *contracting operators*).

Do pierwszej z wymienionych grup należą m.in.:

- jednoargumentowe operatory dokonujące (zazwyczaj nieliniowej) transformacji operandu (w tym przypadku pojedynczej cechy), np. operator logarytmizujący,
- dwu- lub (rzadziej) więcej argumentowe operatory dokonujące agregacji operandu (o rozmiarze $|F'| \geq 2$) do jednej (np. proste wyrażenia arytmetyczne: suma, iloraz, iloczyn, etc.).

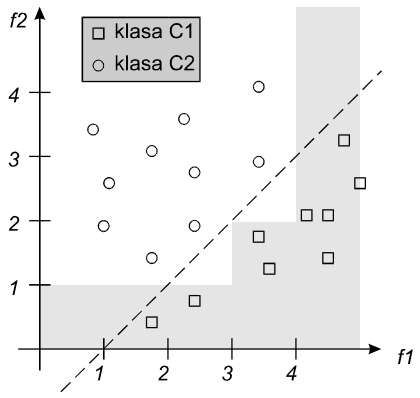
Z kolei przykładami popularnych operatorów zawężających przestrzeń reprezentacji mogą być

- operator usunięcia cechy,
- operator dyskretyzacji cechy. W wyniku jego zastosowania cecha zdefiniowana na skali ciągłej (np. ilorazowej lub przedziałowej) zostaje zastąpiona cechą zdefiniowaną na skali porządkowej (por. zagadnienie dyskretyzacji cech, np. [Fayyad & Irani 1992]).

W rzadziej spotykanych zastosowaniach, np. w uczeniu się klauzul logicznych stosuje się inne typy operatorów, np. operator zamiany stałej na zmienną (np. [Matheus & Rendell 1989]).

Ze zrozumiałych względów, ze zbioru O wyklucza się operatory trywialne, tj. takie, które nie prowadzą do powstania nowej cechy:

$$f_i = o_i(F') \wedge \exists f'_i \in F' : f_i(x) = f'_i(x) \, \forall x \in L$$



Rysunek 4.3: Przykład KI (objaśnienia w tekście)

oraz takie, które dają w wyniku cechy f_i niezależne od obiektów, czyli np. $f_i(x) = const$, $\forall x \in L$[9]. Warto też zaznaczyć, że operator KI powinien być zdefiniowany w taki sposób, aby obliczenie wartości cechy przez niego utworzonej było możliwe dla dowolnych wartości cech wchodzących w skład operandu (a nie tylko tych, które występują w zbiorze uczącym), czyli:

$$\forall o_i \in O,\ f_i = o_i(F') : f_i(x) \neq \phi \forall x \in U$$

Przykład 4 *Załóżmy, że dany jest zbiór L przykładów należących do dwóch klas decyzyjnych C_1 i C_2, opisanych dwoma cechami porządkowymi f_1 i f_2, zilustrowany graficznie na Rys. 4.3.*
Łatwo zauważyć, iż induktor generujący hipotezy w postaci DNF (np. algorytm generowania reguł) potrzebuje co najmniej trzech następujących koniunkcji (reguł) warunków elementarnych (selektorów) nałożonych na poszczególne atrybuty dla opisania klasy decyzyjnej C_1 (odseparowania jej od C_2):
$r1 : (f_1 \geq 4)$
$r2 : (f_2 \leq 1)$
$r3 : (f_1 \geq 3) \wedge (f_2 \geq 2)$
Cieniowane obszary na Rys. 4.3 odpowiadają kombinacjom wartości (f_1, f_2), dla których spełnione są poszczególne reguły. Widać, że ograniczenia języka wyrażania hipotez zastosowanego klasyfikatora regułowego powodują wygenerowanie stosunkowo obszernego opisu, który - choć jest dyskryminujący - nie ujmuje istoty liniowej separowalności klasy C_1 od klasy C_2. Jeżeli natomiast użyć KI z zaledwie jednym

[9] Należy zaznaczyć, że wyeliminowanie trywialnych operatorów nie gwarantuje w ogólności, iż w procesie KI nie pojawią się trywialne cechy.

operatorem konstruktywnym, np. $o(f_i, f_j) = f_i - f_j$*, wówczas możliwe stanie się wygenerowanie nowej cechy* $f_3 = o(f_1, f_2) = f_1 - f_2$ *i do dyskryminującego opisania klasy* C_1 *wystarczy użycie zaledwie jednej reguły* $r1' : (f_3 \geq 1)$*, co stanowi opis bardziej zwięzły i charakteryzujący się lepszymi zdolnościami uogólniania.*

4.4.9 Wybrane metody KI i ich zastosowania

Ponieważ podejście proponowane w niniejszej pracy czerpie jedynie pewną ogólną ideę z konstruktywnej indukcji cech, a także dlatego, że dotyczy ono konstruktywnej indukcji cech obrazu i nie jest metodą względem nich "konkurencyjną", wymienione zostaną tu jedynie najbardziej znane algorytmy uczenia maszynowego stosujące KI. Dokładniejszy przegląd znaleźć można m.in. w pracach [Matheus 1989] i [Wnek & Michalski 1994].

W systemie BACON [Langley, Bradshaw, *et al.* 1983] nowe cechy tworzone są jako proste funkcje cech (wyrażenia arytmetyczne). Selekcja operandu konstruktywnego odbywa się na podstawie analizy zależności pomiędzy cechami w odpowiednio wyselekcjonowanych podzbiorach przykładów. Stosunkowo podobny jest algorytm STAGGER [Schlimmer 1987] z tworzeniem nowych cech jako prostych funkcji logicznych oryginalnych cech dyskretnych i przez dyskretyzację cech ciągłych. W metodzie opisywanej w [Mehra, Rendell, *et al.* 1989] konstrukcja nowych cech odbywa się w *przestrzeni odwrotnej* (ang. *inverted space*). W systemach tych selekcja operandu i operatora bazuje na danych, stąd reprezentują one podejście **DCI**.

Z kolei do ciekawych reprezentantów podejścia **HCI** należą systemy FRINGE [Pagallo 1989] i CITRE [Matheus & Rendell 1989], używające drzew decyzyjnych jako języka reprezentacji hipotez. Bazując na spostrzeżeniu, że w drzewach indukowanych z danych dochodzi często do powtarzania się tych samych warunków elementarnych na różnych ścieżkach od korzenia do liści, generują nowe cechy odpowiadające tak właśnie zidentyfikowanym warunkom.

Stosunkowo wiele rozwiązań można zaliczyć do grupy **KCI**, ponieważ praktycznie każde środowisko uczenia maszynowego umożliwiające definiowanie nowych cech przez użytkownika reprezentuje w pewnym stopniu ten nurt. Historycznie jednym z pierwszych tego typu systemów był AM (ang. *Automated Mathematician*, [Lenat 1977]), który używa predefiniowanych heurystyk dla budowania nowych pojęć w postaci ram (ang. *frames*), tworzenia nowych szczelin (klatek, ang. *slots*) w ramach, oraz ustalania wartości w szczelinach (por. [Mulawka 1996], [Cholewa & Pedrycz 1987]). AM był ponadto zdolny do adaptowania wiedzy przechowywanej w postaci ram do nowych zastosowań. Obecnie jednym z najbardziej znanych systemów wykorzystujących KCI jest algorytm generowania reguł decyzyjnych AQ15 [Michalski, Mozetic, *et al.* 1986], tworzący nowe atrybuty przez stosowanie operatorów logicz-

nych (ang. *l-rules*) i arytmetycznych (ang. *a-rules*).

Literatura nie opisuje zbyt wielu *zastosowań praktycznych KI*. Wydaje się, że do głównych przyczyn takiego stanu rzeczy należy zaliczyć (i) znaczną złożoność obliczeniową wielu metod KI, ograniczającą możliwość ich stosowania do niewielkich (w sensie zbioru oryginalnych cech F_0 i/lub rozmiaru zbioru uczącego L) zbiorów danych, oraz (ii) problemy ze stosowaniem KI dla atrybutów ciągłych, częstych w problemach praktycznych. Zadania, na których weryfikuje się w literaturze techniki KI są zazwyczaj problemami opisanymi cechami dyskretnymi (np. znane problemy MONK1..MONK3 [Thrun, Bala, *et al.* 1991]). Dzieje się tak, ponieważ dla cech dyskretnych dużo łatwiej definiuje się operatory KI.

W pracy [Matheus & Rendell 1989] opisano wykorzystanie KI w uczeniu się rozpoznawania stanów gier planszowych. Z kolei uczenie się (losowo wygenerowanych) funkcji logicznych z elementami KI zaprezentowano w [Matheus & Rendell 1989]. Przydatność wymienionego wyżej systemu CITRE została zweryfikowana doświadczalnie na problemie gry w kółko i krzyżyk [Matheus 1990]. W pracy [Bloedorn & Michalski 1991] zaprezentowano udane zastosowanie algorytmu generowania reguł AQ17 wzbogaconego KI sterowaną danymi (DCI) do rozpoznawania tekstur. Prace [Maloof & Michalski 1997], [Michalski, Rosenfeld, *et al.* 1997] opisują wykorzystanie KI w rozpoznawaniu zapalników w zdjęciach rentgenowskich bagażu lotniczego. W pracy [Krawiec & Słowiński 1997], prezentującej wczesną wersję podejścia opisywanego w niniejszej pracy, zastosowano KI do budowy cech dyskryminujących obrazy mikroskopowe wybranych klas nowotworów ośrodkowego układu nerwowego.

4.4.10 KI a konwencjonalne metody uczenia maszynowego

Konstruktywna indukcja cech jest często stawiana w opozycji do konwencjonalnych metod uczenia maszynowego, określanych w tym kontekście jako metody "selekcyjne" (ang. *selective, non-constructive*). Ścisły podział metod uczenia maszynowego na konwencjonalne "selekcyjne" oraz "konstrukcyjne" jest jednak pewnym nadużyciem, ponieważ w pewnym sensie KI odbywa się w sposób niejawny także w wielu konwencjonalnych systemach uczących się. Na przykład podczas uczenia sztucznych sieci neuronowych neurony w warstwach ukrytych indukują hiperpłaszczyzny w wielowymiarowej przestrzeni cech, tworząc w ten sposób pewne nowe cechy, które agregują w sobie wartości wielu cech obecnych w oryginalnym opisie, często w zawiły, nieliniowy sposób, co stanowi jedną z przyczyn skuteczności tego podejścia. Z kolei w drzewach decyzyjnych z każdym węzłem decyzyjnym skojarzyć można atrybut binarny, którego wartości odpowiadają spełnieniu lub niespełnieniu przez przykład warunku logicznego sformułowanego w tym węźle (jak ma to miejsce w metodach FRINGE [Pagallo 1989] i CITRE [Matheus & Rendell 1989]).

Dla uniknięcia podobnych niejasności należy możliwie precyzyjnie określić, **co stanowi o specyfice KI** na tle konwencjonalnych algorymtów indukcji i innych metod modyfikacji przestrzeni reprezentacji i, w konsekwencji, dla jakich problemów (zadań) stosowanie KI jest najbardziej odpowiednie. Przegląd literatury przedmiotu i zebrane doświadczenie w dziedzinie uczenia maszynowego wskazują na następujące cech decydujących o unikalności KI:

- Możliwość stosunkowo **dowolnego**, ograniczonego jedynie zbiorem użytych operatorów, **modyfikowania** przestrzeni reprezentacji. Cechy indukowane w standardowych klasyfikatorach mają zazwyczaj charakter *statystyczny* (obliczane są w oparciu o rozkłady prawdopodobieństwa cech oryginalnych). KI oferuje mechanizmy bardziej ogólne, umożliwiając m.in. konstrukcję cech o charakterze *relacyjnym*, tj. opisujących pewną relację zachodzącą pomiędzy oryginalnymi cechami [Thornton 1997]. I tak np. znany zbiór przykładów MONK2 [Thrun, Bala, *et al.* 1991] wymaga tak właśnie rozumianej KI, ponieważ do dobrego uogólniania niezbędne jest tu stworzenie nowej cechy, która *zlicza* pewne wartości cech oryginalnych. Innymi słowy, KI stanowi najbardziej ogólną metodę modyfikowania przestrzeni reprezentacji.

- Wyraźne **odseparowanie procesu modyfikacji przestrzeni reprezentacji** od procesu uczenia. W konwencjonalnych algorytmach indukcji transformacje przestrzeni reprezentacji są "wplecione" w działanie algorytmu uczącego i praktycznie nierozerwalnie z nim związane. Zmiany reprezentacji wyznaczone jest zazwyczaj używaną heurystyką i w konsekwencji ekspert (projektant systemu) nie ma na nie bezpośredniego wpływu. Natomiast w KI możliwe jest jawne sterowanie tym procesem (patrz punkt 4.4.7).

Cechy te zdecydowały o wykorzystaniu metodologii KI do transformowania przestrzeni reprezentacji w podejściu zaproponowanym w rozprawie (por. rozdział 5).

4.4.11 Problemy związane z KI

Korzyści wypływające z szerokich możliwości modyfikacji przestrzeni reprezentacji w KI są oczywiste i zostały podkreślone w tej części pracy. Kończąc omawianie KI warto jednak także krótko omówić ważniejsze problemy i otwarte kwestie związane z tym zagadnieniem.

Konstruktywna indukcja boryka się przede wszystkim z pewnymi problemami metodologicznymi, z których najważniejsze to złożoność obliczeniowa i przeuczenie.

1. **Złożoność obliczeniowa**. Wbudowanie procesu uczenia w ramy KI prowadzi w sposób oczywisty do znacznego wzrostu złożoności obliczeniowej procesu uczenia.

Przed przysłowiową eksplozją kombinatoryczną w KI można uchronić się jedynie narzucając ograniczenia na operatory KI i wybierane operandy (podzbiory cech). Wymaga to jednak uwzględnienia dodatkowej wiedzy, której pozyskanie może być kosztowne.

2. **Zwiększone ryzyko przeuczenia**. Ponieważ induktor wyposażony w operatory generujące nowe cechy ma znacznie więcej "stopni swobody" niż induktor pozbawiony tej możliwości (jego przestrzeń hipotez H jest znacznie bardziej rozległa), prawdopodobieństwo przeuczenia jest w obecności KI znacznie większe niż w konwencjonalnych technikach uczenia maszynowego (por. nierówność 4.1, [Kohonen & Sommerfield 1995], [Ng 1998]). Formułując ten problem w kategoriach dylematu ukierunkowanie-elastyczność (por. punkt 4.3), systemy KI charakteryzują się zazwyczaj dużą elastycznością. Stąd kluczem do skutecznej KI jest użycie licznych zbiorów przykładów oraz silnych ukierunkowań indukcyjnych dla zapewnienia dobrej zdolności predykcyjnej wygenerowanej hipotezy.

Ponadto warto wymienić kilka ważniejszych braków i wad KI wynikajacych z kierunków dotychczas prowadzonych badań nad KI oraz ich obecnego stanu opisywanego w literaturze:

1. **Nieprecyzyjne zdefiniowanie.** Jak już zaznaczono wcześniej, nie ma wyraźnej różnicy pomiędzy niektórymi technikami KI a wykształcaniem cech podczas uczenia przez konwencjonalne algorytmy indukcji. Innymi słowy, stosuje się tu milczące założenie o dwuetapowości podejścia (konstrukcja cech + uczenie), kiedy tak naprawdę nie ma różnicy między cechą a konceptem (hipotezą), lub inaczej: nie ma różnicy pomiędzy przestrzenią reprezentacji a przestrzenią hipotez (za: [Fawcett, Gordon, *et al.* 1994]). Można chyba zaryzykować stwierdzenie, że brak formalnego ujęcia KI mogło być jedną z przyczyn zauważalnego spadku zainteresowania tym zagadnieniem w ostatnich latach.

2. **Niewiele zastosowań praktycznych.** Większość prac badawczych prowadzonych w ramach KI stanowią opracowania teoretyczne. Publikowane wyniki testów eksperymentalnych dotyczą zazwyczaj jedynie zbiorów danych stworzonych sztucznie, często z myślą o badaniach dotyczących KI (np. problemy MONK1..MONK3 [Thrun, Bala, *et al.* 1991]). Natomiast mało jest przykładów zastosowań praktycznych (za: [Fawcett, Gordon, *et al.* 1994])

3. **Niewielki przyrost skuteczności.** W eksperymentalnych zastosowaniach różnych technik KI do rzeczywistych problemów obserwuje się zazwyczaj przyrosty zdolności predykcyjnej klasyfikatorów. Jednak są to najczęściej różnice niewielkie (za: [Fawcett, Gordon, *et al.* 1994]).

Podejście proponowane w następnym rozdziale niniejszej pracy zawiera pewne elementy, których celem jest rozwiązanie niektórych z wymienionych tu problemów. W szczególności, przeszukiwanie przestrzeni stanów (reprezentacji) ograniczone jest

do stosunkowo niewielkiego sąsiedztwa bieżącego stanu, co w połączeniu z wstępną selekcją pojedynczych cech (które w pewien sposób odpowiadają ruchom wykonywanym w przestrzeni stanów) prowadzi do znacznego zredukowania złożoności obliczeniowej. Natomiast dla zapobieżenia przeuczeniu, w podejściu *wrapper* stosowany jest odmienny od najczęściej używanej walidacji skrośnej podział na podzbiór uczący i testujący.

Rozdział 5

Opis proponowanego podejścia

W rozdziale tym przedstawiamy proponowane podejście, które wykorzystuje pewne pojęcia i metody uczenia maszynowego i konstruktywnej indukcji cech do wspomagania decyzji na podstawie informacji obrazowej. W szczególności, jak już zaznaczono na początku rozdziału 2, prezentowane podejście przeznaczone jest do rozpoznawania obrazów, jako zadania najbliższego uczeniu się z przykładów i jednocześnie jednego z najbardziej popularnych w zastosowaniach praktycznych. Adaptacja podejścia do innych zadań charakterystycznych dla WDIO jest możliwa, jednak wykraczałaby poza ramy niniejszej rozprawy.

Rozdział otwiera dyskusja przesłanek, którymi kierowano się w trakcie opracowywania podejścia. Dalej następuje omówienie jego ukierunkowania indukcyjnego i analiza złożoności obliczeniowej. Kolejne rozdziały zawierają opisy wykorzystania proponowanego podejścia w zastosowaniach praktycznych.

5.1 Przesłanki

W konwencjonalnym uczeniu maszynowym zakłada się zazwyczaj, że zbiór uczący L, a precyzyjniej wartości cech $f_j(x), f_j \in F$ dla poszczególnych przykładów ze zbioru uczącego $x \in L$, to jedyne źródło informacji o rozważanym problemie dostępne dla systemu uczącego się. Jest to swego rodzaju *założenie zamkniętego świata* (ang. *closed world assumption*). Założenie to, pozornie ograniczające możliwości algorytmu indukcji, w praktyce oznacza realistyczną przesłankę, że wiedza jest zawarta w dostępnych danych. Zapewnia to uniwersalność większości technik uczenia się z przykładów, co jest ich podstawową zaletą i przyczyną wielu udanych aplikacji w różnych dziedzinach. Metody konstruktywnej indukcji cech (KI) podtrzymują w pewnym stopniu to założenie, rozszerzając jednak zakres informacji, z jakich korzystać może algorytm indukcji, o wiedzę zawartą w operatorach KI, która ma charakter

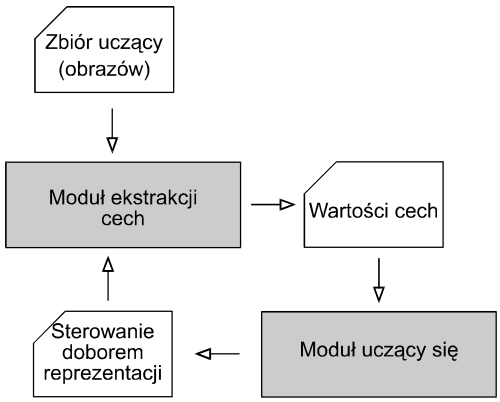
"zdroworozsądkowy" lub wynika z dziedziny zastosowania[1] (por. podrozdz. 4.4.8).

Rozważmy teraz zastosowanie uczenia maszynowego do wnioskowania z informacji obrazowej (WDIO). Gdy przyjmiemy najbardziej popularny dwuetapowy model WDIO (por. rozdział 3 i rys. 3.1), z jednokierunkową komunikacją pomiędzy modułem przetwarzaania i analizy obrazu a modułem uczącym się, ograniczenie się do ustalonej przestrzeni reprezentacji jest raczej naturalne, ponieważ moduł uczący się (algorytm indukcji) bazuje w tym podejściu na opisie obrazu narzuconym przez moduł przetwarzający. Włączenie konstruktywnej indukcji cech **wewnątrz** modułu uczącego się poprawia nieco tę sytuację, ponieważ jest on wówczas zdolny do transformacji przestrzeni reprezentacji wymuszonej przez moduł przetwarzania i analizy. Rozwiązanie takie (por. np. [Maloof & Michalski 1997]) nie prowadzi jednak do postępu o charakterze jakościowym, gdyż punktem wyjścia jest nadal reprezentacja wymuszona przez moduł przetwarzania i analizy, najczęściej wybrana w dość arbitralny sposób przez eksperta dziedziny zastosowania (rys. 3.2). Innymi słowy, takie podejście nie ekploatuje w pełni możliwości wynikających ze specyfiki wnioskowania z informacji obrazowej.

Gdyby jednak pojść za głosem krytyki podejścia dwuetapowego (por. punkt 3.2.1) i dążyć do ściślejszej integracji procesu uczenia w proces przetwarzania i analizy obrazu, sytuacja zmieniłaby się diametralnie. Jeśli rozważy się *użycie KI jako narzędzia wpływania modułu uczącego się na wcześniejsze etapy przetwarzania i analizy obrazu*, zawężenie procesu KI do operowania na cechach wysokiego poziomu dostarczonych przez projektanta systemu (stanowiących pierwotny opis i punkt wyjścia dla procesu KI) przestaje mieć uzasadnienie. Aby uelastycznić ten system, wydaje się rozsądne *przesunąć (rozszerzyć) proces KI na wcześniejsze etapy WDIO (przetwarzanie i analizę obrazu)*. Idea ta, prezentowana w zarysie na rys. 5.1, stanowi trzon proponowanego podejścia.

Formułując tę tezę w inny sposób, wydaje się, że nie ma raczej sensu w sposób sztuczny ograniczać przestrzeni poszukiwań systemu uczącego się do cech zdefiniowanych w sposób subiektywny przez projektanta systemu, nawet gdy wykorzystywana jest KI w konwencjonalnym znaczeniu tego terminu (por. punkt 4.4.2). Pożądane natomiast jest umożliwienie systemowi uczącemu się *wykształcenia cech obrazu*, które nie są jedynie prostymi funkcjami cech eksperta, lecz wnoszą pewną nową jakość do przestrzeni reprezentacji z punktu widzenia WDIO. W pracy [Thornton 1997] wykazano, że KI jest szczególnie przydatna w tych problemach uczenia maszynowego, w których hipoteza musi bazować na nietrywialnej *relacji* wiążącej ze sobą atrybuty warunkowe. Obrazy ze swej natury mają relacyjny charakter (występują w nich m.in. relacje wiążące poszczególne obiekty widoczne w obrazie). Ich uwzględnie-

[1] Nie dotyczy co prawda KI sterowanej wiedzą (KCI, por. punkt 4.4.7), jednak ta metodyka jest stosunkowo rzadko stosowana w praktyce i nie leży w obszarze rozważań niniejszej pracy.



Rysunek 5.1: Ogólny schemat proponowanego podejścia

nie jest często niezbędne dla prawidłowego działania systemu WDIO. Strukturalny charakter danych obrazowych jest zatem dodatkowym argumentem za użyciem konstruktywnej indukcji cech.

Za głębszym wpleceniem procesów uczenia i interpretacji w rozpoznawanie obrazów przemawiają także przesłanki natury biologicznej i psychologicznej. Według tradycyjnych teorii, procesy widzenia i interpretacji obrazu traktowane były jako niezależne, przy czym uważano, iż pierwszy z nich ma charakter bierny, zaś drugi czynny. Nowsze osiągnięcia psychologii poznawczej pokazują jednak, iż nie można oddzielić od siebie procesów widzenia i interpretacji obrazów: *It is no longer possible to separate the process of seeing from that of understanding* ([Zeki 1993], za [Aloimonos, Fermüller, *et al.* 1995]). Obecnie uważa się za pewnik, że rozpoznawanie (zwłaszcza złożonych) obiektów jest *iteracyjnym procesem formułowania hipotez* [Blakemore 1975], [Tomaszewski 1992], w którym faza stawiania hipotezy dotyczącycej identyfikacji obiektu przeplata się z fazą jej weryfikacji. Wynika to z faktu, iż często informacja o postrzeganym obiekcie jest niepełna i/lub nieprecyzyjna. Umysł formułuje hipotezę, którą następnie stara się zweryfikować, przebiegając wzrokiem po obiekcie lub analizując jego chwilowy obraz w pamięci. W kolejnych iteracjach układ wzrokowy zbiera coraz to więcej informacji (np. szczegółów), które mogą być pomocne w potwierdzeniu lub odrzuceniu stawianej hipotezy. Cykl formułowania i weryfikowania hipotez powtarza się aż do znalezienia najbardziej prawdopodobnej interpretacji bodźca, w interesującym nas przypadku - obrazu.

Inne eksperymenty [Schyns, Goldstone, *et al.* 1997] dowodzą z kolei, że *definicje cech wykorzystywanych przez człowieka w trakcie rozpoznawania mogą zmieniać się w trakcie rozpoznawania* pod wpływem procesów poznawczych wysokiego poziomu.

W szczególności, proces tworzenia nowych cech wymaga operowania na stosunkowo niskim poziomie reprezentacji obrazu (bliskim pojedynczym punktom obrazu). Także te wnioski są wykorzystywane w proponowanym podejściu.

Wyżej wymienione spostrzeżenia dotyczące WDIO i przesłanki natury psychofizjologicznej w połączeniu z wadami konwencjonalnych systemów WDIO i problemami związanymi z ich projektowaniem, omówionymi w rozdziałach 2 i 3, wspólnie stanowiły motywację dla rozwinięcia omawianego podejścia. Konstruktywna indukcja cech rozszerzona na etap przetwarzania i analizy obrazu określana będzie dalej terminem *konstruktywnej indukcji cech obrazów*. Następny podrozdział prezentuje to podejście w szczegółach. Wczesna wersja podejścia zaprezentowana była w pracy [Krawiec & Słowiński 1997] pod nazwą LECOMEX (ang. *LEarning COMplex EXamples*) w zastosowaniu do wspomagania diagnozowania nowotworów ośrodkowego układu nerwowego (por. rozdział 7).

5.2 Opis podejścia

Dążąc do ujęcia WDIO w kategoriach uczenia maszynowego, warto wyjść od spostrzeżenia, iż w konwencjonalnym podejściu dwuetapowym obiekt (przykład) x stanowi *opis obrazu* w pewnym języku reprezentacji (wysokiego poziomu), najczęściej w postaci wartości cech $f_j \in F$. Zatem przykład tylko w pewien sposób *odzwierciedla* odpowiadający mu obraz. W odróżnieniu od tej konwencji, w proponowanym podejściu dogodnie jest *utożsamiać* obraz z przykładem x [2]. W takim ujęciu pojedynczej cesze f_j nie odpowiada już, jak to ma miejsce w uczeniu maszynowym, kolumna w tabeli decyzyjnej, lecz pewna *funkcja* o charakterze obliczeniowym, która zastosowana do obrazu x daje w wyniku pewną wartość $f_j(x)$. Stąd w dalszej części pracy terminy "obraz" i "przykład" będą stosowane zamiennie.

W ujęciu funkcjonalnym powyższy zabieg nie wnosi żadnej nowej jakości. Jednak spojrzenie nań z proceduralnego punktu widzenia, charakterystycznego dla informatyki, jako nauki zajmującej się przede wszystkim *metodami* przetwarzania informacji, stawia sprawę w zupełnie nowym świetle. Ważne jest tu spostrzeżenie, iż cecha nie musi być już tylko kolumną w tablicy decyzyjnej lub "czarną skrzynką" generującą wartości dla poszczególnych przykładów (obiektów), lecz pewną *procedurą obliczeniową* (programem) operującą na obrazie x. System można zaprojektować tak, aby procedura ta była wyrażona w języku zrozumiałym dla modułu uczącego się, dzięki czemu będzie miał on dostęp nie tylko do wartości cechy, ale także do jej

[2] Utożsamienie przykładu z obrazem implikuje przydatność proponowanego podejścia głównie w zastosowaniu do *rozpoznawania* obrazów, które jest naturalnym odpowiednikiem nadzorowanej klasyfikacji w uczeniu maszynowym.

opisu (konstrukcji). Opis ten stanowi dodatkowe źródło informacji, które może być wykorzystane przez system uczący się w procesie uczenia. Co więcej, w związku z założeniem o "zamkniętej pętli" w procesie rozpoznawania, wydaje się naturalne, aby podejście to rozszerzyć w ten sposób, by *system uczący miał wpływ na definicję cechy*. W kolejnych podrozdziałach zaprezentowane zostaną dokładniej elementy składowe tej propozycji.

Z punktu widzenia wnioskowania z informacji obrazowej proponowane podejście należy zaklasyfikować jako *wektorowe*, ponieważ bazuje ono na paradygmacie uczenia się z przykładów (rozdział 4) i prowadzi do konstrukcji reprezentacji obrazu w postaci zbioru cech F^+. Z drugiej strony, jak pokażą to dalsze części tego rozdziału, nie oznacza to, iż opisywane podejście ignoruje aspekt strukturalny obrazu.

5.2.1 Podstawowa reprezentacja obrazu

Procedura obliczająca wartość danej cechy operuje na pewnej reprezentacji obrazu, zwanej dalej (*podstawową*) *reprezentacją obrazu*. Oznaczmy przez $r(x)$ podstawową reprezentację obrazu (utożsamianego tu z obiektem/przykładem) x, $x \in U$. W najprostszym przypadku reprezentacja może być równoważna obrazowi ($r(x) = x$), danemu na przykład w postaci mapy bitowej (obraz klasy 1 lub 2 według klasyfikacji Pavlidisa [Pavlidis 1982]). Taka reprezentacja jest jednak zazwyczaj zbyt drobiazgowa i często charakteryzuje się niską przydatnością w zastosowaniach praktycznych, ponieważ pojedyncze punkty obrazu rzadko niosą informację istotną dla zadania rozpoznawania. Z drugiej strony pożądane jest, aby wybrana reprezentacja nie była także sformułowana w kategoriach zbyt abstrakcyjnych (wysokiego poziomu), ponieważ może to zbyt mocno ograniczyć swobodę procesu konstrukcji nowych cech (por. komentarze w przeglądzie literatury, punkt 3.3).

Stąd we WDIO często rozważa się reprezentacje obrazu o pośrednim poziomie abstrakcji, bazujące na pewnych *składowych pierwotnych* (ang. *primitives*), oznaczanych dalej przez a_i. Składowymi pierwotnymi mogą być np. regiony, odcinki, fragmenty krzywych, itp. W proponowanym podejściu składowa pierwotna może być opisana pewnymi cechami (np. współrzędne w obrazie, barwa regionu, długość linii; por. lista poniżej). Cechy te w dalszej części pracy będą nazywane *cechami atomowymi*, dla podkreślenia niemożności ich dekompozycji w ramach podejścia[3]. Ponadto zakłada się, że zbiór wszystkich cech atomowych opisujących składowe pierwotne, oznaczony w dalszej części pracy przez A_0, zadany jest przez projektanta

[3] Składowa pierwotna a jest w ogólności pewną *częścią* podstawowej reprezentacji obrazu $r(x)$. Zapis $a \in r(x)$ jest poprawny jedynie w przypadku, gdy podstawowa reprezentacja obrazu jest *zbiorem* składowych pierwotnych, czego nie zakłada się w ogólnym sformułowaniu podejścia (ma to miejsce w pierwszym z rozważanych zastosowań, patrz rozdział 6).

systemu i pozostaje niezmienny w trakcie całego procesu indukcji cech.

Pojęcie podstawowej reprezentacji obrazu jest na tyle ogólne, że wydaje się, że w jej charakterze można by zatrudnić praktycznie dowolną reprezentację znaną z literatury, w tym na przykład (por. np. [Pavlidis 1987], [Gonzalez & Woods 1992]):

- dla obrazów obiektów dwuwymiarowych:
  - łańcuchy kodów kierunkowych Freemana (ang. *chain codes*),
  - grafy przyległości obszarów (ang. *region adjacency graph*),
  - szkielety (osie środkowe) regionów (ang. *skeleton, medial axis*),
  - reprezentacje wieloskalowe (ang. *multiscale representations*), np. piramidy morfologiczne [Nieniewski 1998].
- dla scen trójwymiarowych:
  - uogólnione cylindry i stożki (ang. *generalized cylinders and cones*).

Wybór podstawowej reprezentacji obrazu zależy od specyfiki zastosowania, na przykład w omawianym w rozdziale 6 zastosowaniu podejścia do rozpoznawania ręcznie pisanych znaków alfanumerycznych za reprezentację obrano punkty należące do szkieletu figury wraz z informacją o lokalnym kierunku.

5.2.2 Konstrukcja cechy

Praktyka przetwarzania i analizy obrazów pokazuje, że do zrealizowania zadania rozpoznawania wymagane jest zazwyczaj przeprowadzenie pewnego ciągu operacji na obrazie źródłowym, z których część ma za zadanie go przetworzyć, a część ekstrahuje z niego pewne cechy. Dlatego wydaje się dogodne dokonać dekompozycji operacji niezbędnych do obliczenia wartości cechy f na etapy przetwarzania. Jeżeli oznaczyć przez $f^{(i)}$ przetwarzanie realizowane w ramach cechy f na i-tym etapie, a przez $|f|$ długość cechy (liczbę etapów przetwarzania), to dekompozycję tę można zapisać formalnie w następujący sposób:

$$\begin{aligned} f &\equiv \left[f^{(1)}, f^{(2)}, \ldots, f^{(|f|)}\right] \\ f(x) &\equiv f^{(|f|)}\left(f^{(|f|-1)}\left(\ldots f^{(2)}\left(f^{(1)}\left(r\left(x\right)\right)\right)\ldots\right)\right), \quad x \in U \end{aligned} \tag{5.1}$$

przy czym dziedziną funkcji realizującej pierwszy etap przetwarzania $f^{(1)}$ jest podstawowa reprezentacja obrazu r (patrz poprzedni podrozdział), zaś ostatni etap przetwarzania $f^{(|f|)}$ daje w wyniku wartości skalarne (np. liczbowe lub nominalne).

Ideę tę schematycznie prezentuje rysunek 5.2. Poza tym wymagana jest odpowiedniość przeciwdziedziny i dziedziny odpowiednio i-tego i $i+1$-go etapu przetwarzania:

$$D^{-1}\left(f^{(i)}\right) \subseteq D\left(f^{(i+1)}\right),\ i = 1, \ldots, |f| - 1$$

W obecnej wersji proponowanego podejścia zakłada się, że wartości cech są liczbami rzeczywistymi (uogólnienie na cechy określone na skali nominalnej jest możliwe, jednak w rozważanych zastosowaniach było raczej zbyteczne):

$$D^{-1}\left(f^{(|f|)}\right) \subseteq \Re$$

Przez pewną analogię sposobu konstruowania cechy w proponowanym podejściu do indukowania nowych cech w konstruktywnej indukcji, $f^{(i)}$ będziemy nazywać krótko *operatorem*, a przez O oznaczymy zbiór wszystkich rozważanych operatorów konstruktywnej indukcji cech obrazów[4] (por. punkt 4.4.8).

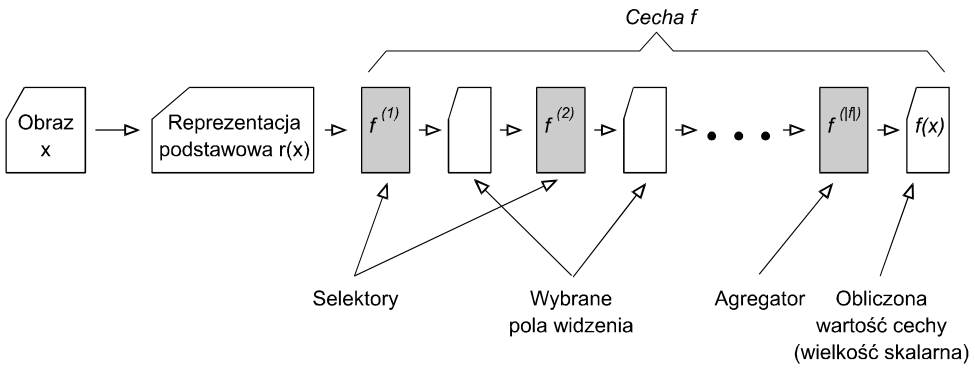
5.2.3 Operatory $f^{(i)}$

Cel, jakim jest rozpoznawanie obrazu, charakterystyka danych wejściowych (podstawowa reprezentacja obrazu) oraz fakt, iż ostateczne wartości cechy muszą być skalarne, narzucają pewne ograniczenia na sposób przetwarzania realizowany przez cechę f. Ogólną tendencją powinna być *redukcja informacji* na poszczególnych etapach przetwarzania, która jest niezbędna do tego, aby znaczną ilość informacji wejściowej (podstawowa reprezentacja obrazu $r(x)$) scharakteryzować ostatecznie w postaci skalarnej wartości cechy. W proponowanym podejściu redukcję tę uzyskuje się poprzez *selekcję* i *agregację* informacji. Operatory pierwszego typu zawężają pole widzenia do wybranego fragmentu obrazu, zaś drugiego typu - obliczają na jego podstawie wielkość skalarną, czyli końcową wartość cechy $f(x)$. Choć takie rozwiązanie nie wyczerpuje wszystkich możliwości, zostało przyjęte w opisywanym podejściu jako proste i jednocześnie nietrywialne. W kolejnych podrozdziałach omówione zostaną operatory reprezentujące obie te grupy.

Operatory selekcji pola widzenia

Nietrywialne zastosowania rozpoznawania obrazów prawie zawsze dotyczą analizy obiektów złożonych. Stąd wiele technik rozpoznawania dokonuje (w sposób jawny badź niejawny) dekompozycji obiektów, i zawęża zakres danych wejściowych do wybranych fragmentów analizowanego obrazu, definiując tzw. *pole widzenia* (ang. *region of interest, ROI*), oznaczone dalej przez $R(x)$.

[4]Warto zaznaczyć, iż w szczególnym przypadku kolejność stosowania operatorów $f^{(i)}$ w ramach cechy f może nie grać roli; nie jest to jednak regułą.



Rysunek 5.2: Sposób obliczania pojedynczej cechy obrazu w proponowanym podejściu

Proponowane podejście umożliwia włączenie selekcji pola widzenia jako jednego z etapów przetwarzania $f^{(i)}$. Operator selekcji pola widzenia będziemy nazywać krótko *operatorem selekcji* lub *selektorem*, a symbolem S, $S \subset O$ będziemy oznaczać zbiór wszystkich rozważanych selektorów. Zgodnie z powyższymi uwagami selektor implementuje operację *antyekstensywną*, tj. dla dowolnego pola widzenia R i dowolnego selektora $s \in S$ zachodzi $s\,(R) \subseteq R$.

Sposób działania operatora selekcji ROI zależy przede wszystkim od przyjętej podstawowej reprezentacji obrazu $r(x)$. Na przykład dla reprezentacji podstawowej w postaci mapy bitowej (obrazu rastrowego) selekcja ta może polegać na ograniczeniu analizy obrazu do pewnego okna, co odpowiada na przykład koncepcji pól recepcyjnych (ang. *receptive fields*) w niektórych typach sztucznych sieci neuronowych [Fukushima 1975], [Fukushima 1980], [LeCun & Bengio 1995]. Gdy z kolei podstawową reprezentacją obrazu jest graf przyległości obszarów, selektor może wybierać pewien fragment grafu (np. pojedynczy węzeł wraz z sąsiadami) do dalszej analizy.

Warto zaznaczyć, że w ogólności selekcja pola widzenia nie musi być przeprowadzana zgodnie z kryteriami bliskości topologicznej, choć jest to zgodne z charakterystyką percepcji wzrokowej i w praktyce najczęściej przydatne. Selektor może kierować się w tym procesie innymi wielkościami, tj. innymi niż współrzędne przestrzenne cechami atomowymi opisującymi składowe pierwotne (por. operatory opisywane w rozdziale 6). W konsekwencji pole widzenia w proponowanym podejściu jest pojęciem bardziej ogólnym niż to się konwencjonalnie rozumie w rozpoznawaniu obrazów, tj. nie musi stanowić spójnego przestrzennie fragmentu podstawowej reprezentacji obrazu $r\,(x)$.

Operatory agregacji cech pola widzenia

Zadaniem operatora agregacji cech pola widzenia jest obliczenie końcowej wartości cechy $f(x)$ na podstawie składowych pierwotnych obecnych w polu widzenia $R(x)$ wyselekcjonowanym uprzednio przez sekwencję selektorów. W związku z tym agregacja stanowi zawsze ostatni etap przetwarzania realizowanego w ramach cechy f, czyli $f^{(|f|)}$ (wyrażenie 5.1) i występuje w niej dokładnie raz (rys. 5.2).

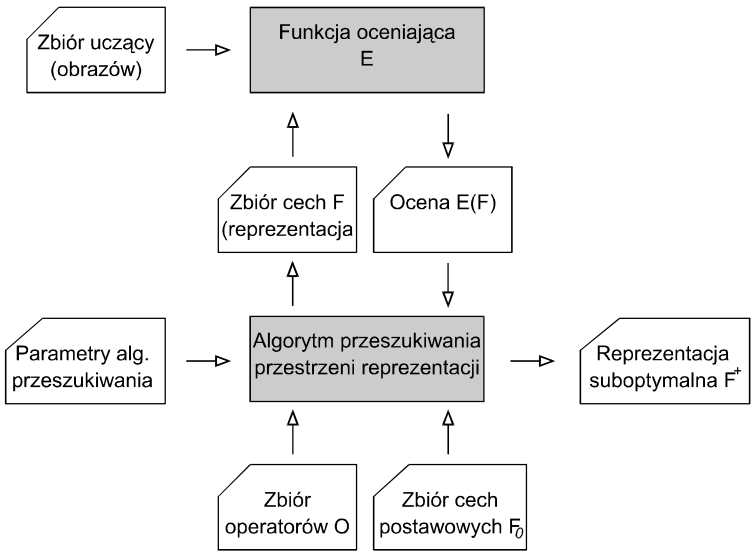
W najprostszym przypadku operator agregujący może statystycznie opisywać składowe pierwotne, obliczając na przykład ilość składowych pierwotnych obecnych w polu widzenia $R(x)$. W przypadku bardziej wyrafinowanym operator agregujący może być pewną funkcją wartości cech atomowych składowych pierwotnych. W wielu zastosowaniach praktycznych przydatne będzie tu wykorzystanie agregacji charakterystycznych dla statystyki opisowej, na przykład średniej arytmetycznej, zakresu, wariancji, wybranej cechy atomowej z A_0. Operator agregacji pola widzenia będziemy nazywać krótko *operatorem agregacji* lub *agregatorem*, a symbolem $A \subset O$ będziemy oznaczać zbiór wszystkich rozważanych agregatorów. Z definicji $A \cap S = \emptyset$ i $A \cup S = O$.

Operatory a cechy atomowe

Wynik zastosowania operatora (selekcji lub agregacji) jest w ogólności zależny od jego charakterystyki oraz zawartości podstawowej reprezentacji obrazu, a dokładniej od wartości cech atomowych A_0 składowych pierwotnych a_i obecnych w bieżącym polu widzenia $R(x)$. W proponowanym podejściu przyjęto upraszczające założenie, że dany operator $o \in O$ bierze pod uwagę wartości *jednej* wybranej cechy atomowej a_i. Jednak, jak to zostaje pokazane w kolejnych punktach i w opisie zastosowań, wiele operatorów (np. minimum i maksimum) ma uniwersalny charakter i może być stosowanych do różnych cech atomowych. W praktyce projektant systemu nie musi zatem specyfikować wszystkich operatorów w O *explicite*; wystarczy w tym celu zdefiniować operatory "uniwersalne" (tj. takie, dla których cecha atomowa a_i będąca podstawą obliczeń nie jest jeszcze znana) i sprecyzować, które cechy atomowe mogą być przezeń wykorzystywane. Takie rozwiązanie przyjęto w implementacji komputerowej podejścia i przy prezentowaniu zastosowań praktycznych w kolejnych rozdziałach.

5.2.4 KI w proponowanym podejściu

Jednym z kluczowych punktów niniejszej pracy jest ujęcie procesu konstrukcji cechy obrazu w kategoriach konstruktywnej indukcji cech. W szczególności pojedynczy etap przetwarzania $f^{(i)}$ odpowiada operatorowi konstruktywnemu, a jego dziedzina



Rysunek 5.3: Schemat proponowanego podejścia

$D(f^{(i)})$ operandowi (por. punkt 4.4). Zasadnicza odmienność proponowanego podejścia od konwencjonalnej KI polega na tym, że operand nie jest wektorem wielkości skalarnych (por. punkt 4.4.3), lecz polem widzenia (najczęściej zbiorem). Realizując sformułowany na początku tego rozdziału postulat o konieczności wpływu modułu uczącego się na przestrzeń reprezentacji, proponowne podejście zatrudnia metodykę KI i selekcji cech do modyfikowania cechy reprezentacji (zbioru cech) F oraz pojedynczych cech f_j. W szczególnośći modyfikacje dotyczące selektorów prowadzą do zmiany sposobu określania pola widzenia wykorzystywanego przez daną cechę, zaś agregatorów do zmiany wielkości wyznaczanej ze składowych pierwotnych obecnych w polu widzenia R.

Ponieważ opisywane podejście łączy w sobie elementy KI, selekcji cech i rozpoznawania obrazów, trudno dopatrzyć się w nim wyraźnego podziału na fazy KI opisywane w punkcie 4.4.3, natomiast dogodnie jest opisać je w kategoriach *przeszukiwania dyskretnej przestrzeni stanów* (por. punkt 4.4.5).

Jak to zaznaczono na rysunku 5.3, prezentującym ogólny schemat podejścia, algorytm przeszukiwania przestrzeni rozwiązań F jest wyraźnie wydzielonym komponentem systemu konstruktywnej indukcji cech obrazu, co umożliwia stosowanie praktycznie dowolnego algorytmu spełniającego pewne wymienione niżej ograniczenia.

W związku ze wspomnianą w punkcie 4.4.5 wykładniczą złożonością pełnego przeszukiwania przestrzeni stanów, dla uzyskania wyników w rozsądnym czasie nie-

zbędne jest wykorzystanie podejść przybliżonych (heurystyk). W konsekwencji w miejsce rozwiązania (stanu) optymalnego F^* gwarantowanego w algorytmach dokładnych otrzymujemy rozwiązanie F^+ będące pewnym *suboptimum* (optimum lokalnym), tj. $E(F^+) \leq E(F^*)$. W ogólności rozważać tu można zastosowanie dowolnej *metaheurystyki* nadającej się do przeglądania dyskretnej przestrzeni stanów, np. przeszukiwania lokalnego, przeszukiwania tabu [Laguna, Barnes, *et al.* 1991], symulowanego wyżarzania [van Laarhoven & Aarts 1987], algorytmów genetycznych (ewolucyjnych) [Von Neuman 1966], [Holland 1975], [Michalewicz 1996]. Niezależnie od wybranego algorytmu, niezbędne jest tu określenie następujących elementów:

1. Sposobu modyfikowania bieżącego rozwiązania (*ruchów*) i ograniczeń nałożonych na ich dopuszczalność,
2. Sposobu (algorytmu) przeglądania przestrzeni rozwiązań F,
3. Funkcji oceniającej E.

Ad 1) Ze względu na to, że pojedyncze rozwiązanie F_k jest zbiorem cech, z których każda jest ciągiem operatorów, ruchy dzieli się w proponowanym podejściu na dwie kategorie:

- ruchy dotyczące całych cech (realizujące zasadniczo proces selekcji cech, por. punkt 4.4.6),
- ruchy dotyczące poszczególnych operatorów KI (w wybranej cesze z F_k).

Ruchy dotyczące całych cech prowadzą najczęściej do rozwiązań różniących się znacząco od rozwiązania bieżącego, podczas gdy ruchy dotyczące poszczególnych operatorów wprowadzają stosunkowo niewielkie zmiany w bieżącym rozwiązaniu. Algorytm przeszukujący dysponuje dzięki temu zarówno możliwością szybkiego przemieszczania się w przestrzeni rozwiązań, jak i precyzyjnego ulepszania bieżącego rozwiązania. Zbiory ruchów należących do poszczególnych kategorii określane są na podstawie zawartości dwóch zbiorów, stanowiących parametr metody i ustalanych przez projektanta:

- zbioru *cech podstawowych* F_0,[5]
- zbioru operatorów KI O.

[5]Kierując się podobną interpretacją, przyjęto tutaj oznaczenie F_0, stosowane w punkcie 4.4.6 na oznaczenie oryginalnego zbioru cech w problemie selekcji.

Precyzyjna interpretacja zawartości tych zbiorów podana jest w dalszych punktach tego rozdziału.

Ad 2) W ramach niniejszej pracy zdecydowano się na empiryczną weryfikację (por. rozdziały 6 i 7) pewnej odmiany algorytmu przeszukiwania lokalnego (przeszukiwanie strome) oraz metaheurystyki algorytmów ewolucyjnych, implementującej przeszukiwanie globalne. Kolejne punkty prezentują algorytmy przeszukiwania oraz sposób reprezentowania i modyfikowania rozwiązania dla obu tych metaheurystyk.

Ad 3) Funkcja oceniająca, jako element niezależny od stosowanego algorytmu, opisana jest w osobnym punkcie.

Strome przeszukiwanie lokalne (SLS)

Algorytm lokalnego przeszukiwania stromego (ang. *steep local search,* SLS, *hill climbing*) został zastosowany ze względu na prostotę i brak parametrów. Lokalny charakter przeszukiwania implikuje tu konieczność określenia dopuszczalnych możliwości modyfikowania bieżącego rozwiązania, czyli *ruchów* przeprowadzających rozwiązanie bieżące (stan bieżący) w inne rozwiązanie (stan).

Modyfikowanie bieżącego rozwiązania Dla uproszczenia zapisu, oznaczmy bieżące rozwiązanie F_k (stan, podzbiór cech) przez F, zaś stan zmodyfikowany, do którego prowadzi rozważany ruch, przez F'. Lokalny charakter przeszukiwania oznacza, iż stan F' musi być w pewnym stopniu *podobny* do stanu bieżącego F. W proponowanym podejściu uzyskuje się to przez rozważanie ograniczonego zbioru ruchów zaprezentowanego poniżej. Dzięki temu wykonywanie kolejnych ruchów powoduje *stopniową* modyfikację rozwiązania, co zapewnia ciągłość procesu przeszukiwania. Ograniczenie liczby potencjalnych ruchów redukuje ponadto złożoność obliczeniową podejścia.

Jak to już zaznaczono w punkcie 5.2.4, ruchy podzielone są na dwie kategorie. **Ruchy dotyczące całych cech** polegają na wykonaniu jednej z dwóch następujących operacji:

- *dodanie* (wstawienie, ang. *insertion*) cechy f:

$$F \xrightarrow{Ins(f)} F' = F \cup \{f\}\,,\ f \in F_0$$

- *usunięcie* (ang. *deletion*) cechy f:

$$F \xrightarrow{Del(f)} F' = F \setminus \{f\}\,,\ f \in F$$

gdzie F_0 to zbiór cech podstawowych, stanowiący parametr metody i ustalany przez użytkownika[6].

Z kolei **ruchy dotyczące poszczególnych operatorów** KI modyfikują wybraną cechę należącą do bieżącego stanu $f \in F$, $f = \left[f^{(1)}, ..., f^{(|f|)}\right]$. Zatem w ogólności wynikiem wykonania ruchu tego typu jest przejście do stanu $F' = (F \setminus \{f\}) \cup \{f'\}$, gdzie f' jest cechą uzyskaną przez pewną modyfikację cechy f. Postać f' zależy od wykonywanej operacji, którą może być:

- *wstawienie* operatora $o \in O$ na i-tej pozycji w cesze f:

$$F \xrightarrow{InsOp(f,o,i)} F'$$
$$gdzie\ f' = \left[f^{(1)}, ..., f^{(i-1)}, o, f^{(i)}, ..., f^{(|f|)}\right],$$

- *usunięcie* operatora występującego na i-tej pozycji w cesze f:

$$F \xrightarrow{DelOp(f,i)} F'$$
$$gdzie\ f' = \left[f^{(1)}, ..., f^{(i-1)}, f^{(i+1)}, ..., f^{(|f|)}\right],$$

- *zastąpienie* operatora występującego na i-tej pozycji operatorem $o \in O$ w cesze f:

$$F \xrightarrow{RepOp(f,o,i)} F'$$
$$gdzie\ f' = \left[f^{(1)}, ..., f^{(i-1)}, o, f^{(i+1)}, ..., f^{(|f|)}\right],$$

gdzie O to zbiór operatorów KI zdefiniowany przez użytkownika, $i = 1, ..., |f|$.

Powyższe punkty definiują wszystkie ruchy, jakie potencjalnie mogą być wykonywane przy wykorzystaniu cech ze zbioru F_0 i operatorów ze zbioru O. Jednak wykonywanie niektórych z nich może prowadzić do stanów, które są niepożądane z punktu widzenia efektywności procesu przeszukiwania lub zawierają cechy nie mające poprawnej interpretacji. Zbiór rozważanych ruchów należy zatem zawęzić, co prowadzi do koncepcji ruchu dopuszczalnego.

Ruch λ prowadzący ze stanu F do stanu F' ($F \xrightarrow{\lambda} F'$) jest *dopuszczalny*, jeżeli F' spełnia pewne ograniczenia w stosunku do F, co zapisujemy $Constr\,(F', F)$. W proponowanym podejściu stosowane są dwie grupy ograniczeń:

[6] Ponadto rozważano wykorzystanie operatora *zastępowania* (wymiany) cechy z F. Jego wykorzystanie jest jednak obarczone znaczną złożonością obliczeniową, ponieważ w odróżnieniu od ruchów *Ins* i *Del* wybraną cechą z F_N można zastąpić *jedną z* cech w F. Ponadto z funkcjonalnego punktu widzenia zastąpienie cechy da się zrealizować przy pomocy pary ruchów *Ins* i *Del*. Stąd ostatecznie zrezygnowano z zastępowania cech.

- ograniczenia związane z procesem przeszukiwania przestrzeni stanów:
  - $F' \neq F$ (zapobieganie ruchom pustym), i
  - $\forall f_1, f_2 \in F'$, $f_1 \neq f_2$ (zapewnienie unikalności cech w rozwiązaniu), i
  - $|F'| \leq |F|_{\max}$, gdzie $|F|_{\max}$ to maksymalna liczność rozwiązania (zapobieganie nadmiernemu rozrostowi wyindukowanego zbioru cech), i
  - $\forall g \in F' : |f| \leq |f|_{\max}$, gdzie $|f|_{\max}$ to maksymalna długość cechy (zapobiega przespecjalizowaniu wygenerowanych cech),
- ograniczenia dotyczące zapewnienia poprawności składniowej cechy:
  - jeśli $\lambda = DelOp \Rightarrow i < |f|$ (usuwanie agregatora jest niedopuszczalne), i
  - jeśli $\lambda = RepOp \Rightarrow (f \in S \Rightarrow f' \in S) \wedge (f \in A \Rightarrow f' \in A)$ (selektor może być zastępowany tylko selektorem, zaś agregator tylko agregatorem).

Sąsiedztwem $N(F)$ *stanu* F nazywamy zbiór stanów F', do których można wykonać *dopuszczalny* ruch ze stanu F. Ze względu na występowanie w proponowanym podejściu dwóch typów ruchów, sąsiedztwo warto podzielić na:

- sąsiedztwo ze względu na ruchy dotyczące całych cech:

$$N_{F_0}(F) = \{F' : F \xrightarrow{Ins(f')} F' \vee F \xrightarrow{Del(f)} F' \vee F \xrightarrow{Rep(f,f')} F', \\ f \in F, f' \in F_0, Constr(F', F)\}$$

- sąsiedztwo ze względu na ruchy dotyczące operatorów KI:

$$N_O(F) = \{F' : F \xrightarrow{InsOp(f,o,i)} F' \vee F \xrightarrow{DelOp(f,i)} F' \vee F \xrightarrow{RepOp(f,o,i)} F', \\ f \in F, o \in O, i \in \langle 1, |f| \rangle, Constr(F', F)\} \tag{5.2}$$

Wówczas:

$$N(F) = N_{F_0}(F) \cup N_O(F) \tag{5.3}$$

Podsumowując należy zaznaczyć, iż specyfiką proponowanego podejścia jest to, iż liczność sąsiedztwa $N(F)$ danego rozwiązania F silnie zależy od rozmiaru rozwiązania $|F|$, co ma pewne implikacje dla procesu przeszukiwania przestrzeni rozwiązań $\mathcal{F}$ w algorytmie SLS.

Algorytm SLS Algorytm 5.2.1 prezentuje heurystykę stromego przeszukiwania lokalnego *SLS*. Algorytm rozpoczyna działanie od stanu $F = \varnothing$ i przechodzi zawsze do tego stanu w sąsiedztwie, który maksymalizuje przyrost wartości E. Po wykonaniu każdego ruchu kontrolowana jest minimalność (w sensie liczby cech) bieżącego rozwiązanie F (eliminowane z F są cechy, których usuwanie nie prowadzi do spadku $E(F)$). Warunkiem zatrzymania algorytmu jest brak możliwości poprawy bieżącego rozwiązania F poprzez wykonanie któregoś z dozwolonych ruchów (odpowiednik fazy detekcji w schemacie KI prezentowanym w podrozdziale 4.4.3).

Algorytm 5.2.1 ***SLS***

begin
  $F \leftarrow \varnothing$
  loop
    Stosując ruchy $\{Ins, Del, InsOp, DelOp, RepOp\}$
      utwórz sąsiedztwo $N(F)$ *stanu* F
    Wybierz najlepszy stan z sąsiedztwa $F' \in N(F)$*:*
        $F' = \arg \max_{F_i \in N(F)} E(F_i)$
    if $E(F') \leq E(F)$ ***then***
      return F
    endif
    $F \leftarrow F'$
    foreach $f \in F$
      if $E(F \setminus \{f\}) \geq E(F)$ ***then***
        $F \leftarrow F \setminus \{f\}$
      endif
    endfor
  endloop
end

Poza podstawową wersją algorytmu SLS w niniejszej pracy rozważa się także pewne jej modyfikacje, dotyczące dwóch niezależnych aspektów:

1. Preferowania pewnych grup ruchów,
2. Zapobiegania utknięciu przeszukiwania w lokalnych minimach.

Ad.1) W punkcie 5.2.4 zaprezentowano dwa typy ruchów wykorzystywanych wraz z algorytmem SLS: operujących na całych cechach oraz operujących na pojedynczych operatorach (wzór 5.3). Poza oczywistym podejściem polegającym na traktowaniu ruchów reprezentujących obie te grupy jednakowo nasuwają się dwie inne strategie:

- preferowanie ruchów dotyczących całych cech (SLS^f),
- preferowanie ruchów dotyczących operatorów (SLS^c).

Aby nie komplikować zbytnio podejścia, preferencje te zostały zaimplementowane w najprostszy możliwy sposób ("leksykograficzny"), tj. ruchy niepreferowane rozważane są dopiero wówczas, gdy zbiór ruchów preferowanych polepszających rozwiązanie jest pusty. Użycie pierwszej z wymienionych opcji prowadzi do szybkiego wykształcenia przez algorytm przeszukujący "zgrubnego" i stosunkowo licznego (w sensie liczby cech) rozwiązania. Z kolei w drugim podejściu rozważa się dodanie lub usunięcie cechy dopiero wówczas, gdy bieżącego rozwiązania nie da się już lepiej "dostroić" przez manewrownie operatorami KI.

Ad.2) W prostej postaci algorytmu SLS brak jest mechanizmu zapobiegającego utknięciu procesu przeszukiwania w lokalnym minimum. SLS jest w związku z tym w ogólności bardzo podatny na ten problem, w tym także w przypadku zastosowania go w KI (por. wyniki w rozdziale 6). Dlatego niezbędne staje się wyposażenie go w taką możliwość. W prowadzonych badań rozważono dwie grupy metod zapobiegania problemowi minimów lokalnych: uniwersalne i dostosowane do przeszukiwania przestrzeni zbiorów cech w KI.

W ramach pierwszej grupy zastosowano nawroty (ang. *backtracking*). W metodzie tej, jeżeli dla bieżącego stanu nie istnieje ruch dopuszczalny polepszający wartość funkcji E, następuje powrót do poprzednio odwiedzonego stanu. Jednocześnie tworzy się i aktualizuje listę odwiedzonych stanów aby zapobiec zapętleniu procesu przeszukiwania. Oznaczmy tę heurystykę przez SLS_B.

W ramach drugiej grupy wykorzystuje się wiedzę dziedzinową, tj. fakt, że rozwiązanie jest zbiorem cech, i dokonuje jego nieznacznej modyfikacji. Możliwe modyfikacje można w ogólności podzielić na rozbudowujące bieżące rozwiązanie (stan), np. przez dodanie cechy lub operatora KI, oraz na modyfikacje zawężające (upraszczające) bieżące rozwiązanie, np. poprzez usunięcie cechy lub operatora KI. Kierując się sprawdzonym w uczeniu maszynowym ukierunkowaniem na hipotezy proste, w proponowanej metodzie rozważano jedynie drugą z wymienionych tu metod.

W szczególności rozważano dwie proste metody modyfikacji bieżącego rozwiązania:

- usuwanie najmniej informatywnej cechy (SLS_L),
  tj. cechy $f' = \max \arg_{f \in F} E\left(F \backslash \{f\}\right)$,
- usuwanie najbardziej informatywnej cechy (SLS_M),
  tj. cechy $f' = \min \arg_{f \in F} E\left(F \backslash \{f\}\right)$.

Te ekstremalne względem siebie podejścia charakteryzują się komplementarnymi cechami. SLS_L dokonuje mniej istotnej modyfikacji stanu niż SLS_M, co z jednej strony zapewnia, że główny "trzon" rozwiązania zostaje zachowany, z drugiej jednak strony tak niewielki ruch w przestrzeni stanów może nie zapewnić opuszczenia niecki lokalnego minimum. Eksperymenty obliczeniowe opisywane w rozdziale 6 dotyczyły m.in. zweryfikowania trzech wymienionych tu sposobów zapobiegania utknięciu w minimum lokalnym.

Algorytm ewolucyjny (AE)

Jako heurystykę przeszukiwania przestrzeni rozwiązań alternatywną względem wybierania wstępującego zastosowano algorytm ewolucyjny [Holland 1975], [Goldberg 1989], [Michalewicz 1996]. Za wyborem tej metaheurystyki przemawiały zdolność do radzenia sobie z lokalnymi optimami funkcji oceniającej E, równoległy charakter procesu przeszukiwania przestrzeni rozwiązań oraz skuteczność wykazywana w wielu zastosowaniach praktycznych, w tym w selekcji cech (por. np. [Vafaie & Imam 1994], [Yang & Honavar 1998], [Komosiński & Krawiec 2000]).

Kodowanie i modyfikowanie rozwiązań Kluczową kwestią przy wykorzystywaniu algorytmu ewolucyjnego jest sposób reprezentowania rozwiązań w postaci osobników, czyli *ciągów kodowych* ("chromosomów"). W proponowanym podejściu przyjęto najprostsze możliwe rozwiązanie, utożsamiając osobnika ze *zbiorem* cech F. Rozwiązanie to wydaje się eleganckie, jest jednak nietypowe z punktu widzenia algorytmów genetycznych, gdzie ciągi kodowe są *wektorami* liczb (tzw. *alleli*). Implikuje to konieczność zaprojektowania specjalizowanych operatorów rekombinacji (krzyżowania i mutacji), w których pozycja elementu rozwiązania w ciągu kodowym nie odgrywa znaczenia[7]. Wykorzystywany tu algorytm trafniej jest zatem określać algorytmem ewolucyjnym niż genetycznym, z racji wykorzystywania w sposobie kodowania rozwiązań i w operatorach rekombinacji wiedzy specyficznej dla problemu (por. [Michalewicz 1996], s. 27). Co więcej, ponieważ zbiór cech realizuje w gruncie rzeczy pewien *program* przetwarzania i analizy obrazu, wersja proponowanego podejścia wykorzystująca algorytmy ewolucyjny w interesujący sposób upodabnia się w pewnym stopniu do koncepcji *programowania genetycznego* [Koza 1994].

Operacja krzyżowania przeprowadzana dla pary osobników (F_1, F_2) zdefiniowana jest podobnie jak w konwencjonalnym algorytmie ewolucyjnym i polega na wymianie losowo wybranych komponentów rozwiązań, za które przyjęto cechy. Taka

[7] Z teoretycznego punktu widzenia zbiory można kodować w postaci ciągów binarnych (zero-jedynkowych), przy bardzo dużej liczbie potencjalnych elementów takiego zbioru jest to jednak nierealne ze względu na złożoność pamięciową.

definicja ma tę zaletę, że praktycznie daje zawsze w wyniku dopuszczalne rozwiązania potomne, tak więc nie ma potrzeby poźniejszego ich "naprawiania" (co mogłoby być procesem kosztownym obliczeniowo, por. [Michalewicz 1996] s. 219). Jedynym wyjątkiem od tej reguły jest możliwość pojawienia się duplikatów cech w rozwiązaniach potomnych F_1 i F_2, jednak ich usunięcie jest bardzo prostą operacją[8].

Operacja mutacji przeprowadzana dla pojedynczego osobnika F polega na modyfikacji losowo wybranego operatora w losowo wybranej cesze $f \in F$. Innymi słowy, modyfikacja polega na wykonaniu ruchu do losowo wybranego stanu w sąsiedztwie $N_O(F)$ (wzór 5.2) składającym się z rozwiązań utworzonych z F przez operacje na pojedynczych operatorach KI ze zbioru O zadanego przez projektanta (por. punkt 5.2.4).

Krzyżowanie w proponowanym podejściu przebiega zatem na całych cechach, podczas gdy operacja mutacji dotyczy pojedynczych operatorów KI, co dobrze odpowiada metodyce algorytmów ewolucyjnych, gdzie mutacja powinna polegać na minimalnej modyfikacji rozwiązania (osobnika).

Algorytm AE Zastosowano tradycyjną postać algorytmu ewolucyjnego (algorytm 5.2.2, por. np. [Michalewicz 1996]). Zgodnie z przyjętą w środowisku algorytmów ewolucyjnych terminologią, pojedyncze rozwiązanie jest tu nazywane *osobnikiem*, a funkcja E funkcją *przystosowania*, ang. *fitness function*). Faza inicjalizacji populacji osobników P polega na stworzeniu zadanej liczby $|P|_0$, z których każdy składa się z ustalonej liczby cech $|F|_0$, wylosowanych z równomiernym rozkładem prawdopodobieństwa ze zbioru cech podstawowych F_0 (por. punkt 5.2.4). Selekcja (ewolucyjna) przebiega zgodnie z tradycyjną regułą (tzw. regułą koła ruletki), tj. osobniki do rekombinacji wybierane są z prawdopodobieństwem proporcjonalnym do ich przystosowania, tj. prawdopodobieństwo wyboru osobnika F w populacji P wynosi $E(F) / \sum_{F' \in P} E(F')$. Krzyżowanie i mutacja realizowane są w sposób opisany w poprzednim punkcie.

[8] Warto zaznaczyć, że eliminowanie duplikatów cech jest przeprowadzane głównie dla minimalizacji rozmiaru reprezentacji $|F^+|$, z punktu widzenia funkcji oceniającej (przystosowania) E typu *wrapper* nie ma zasadniczo przeciwskazań dla używania wielu takich samych cech w opisie. Co więcej, niektóre eksperytmenty obliczeniowe opisywane w rozdziale 6 pokazywały, że powielenie cechy w opisie może prowadzić do wzrostu wartości E; dotyczyło to głownie podejścia *wrapper* wykorzystującego kNN jako klasyfikator, gdzie cecha występująca wielokrotnie staje się "ważniejsza".

Algorytm 5.2.2 ***AE***

begin
Przeprowadź inicjalizację populacji P
$F \leftarrow \emptyset$ *- najlepszy osobnik*
loop
Wygeneruj nową populację P' *przez selekcję par osobników z* P
i ich krzyżowanie
$P' \leftarrow P$
Przeprowadź mutację wybranych osobników z P
(z prawdopowobieństwem p_{mut}*),*
Przeprowadź lokalną optymalizację osobników z P
Oceń osobników z P *funkcją przystosowania* E
Znajdź najlepszego osobnika w bieżącej populacji $F' = \arg\max_{F_i \in P} E(F_i)$
if $E(F') > E(F)$ ***then***
$F \leftarrow F'$
endif
while *nie osiągnięto maksymalnej (zadanej) liczby iteracji*
return F
end

Jak wspomniano już wcześniej, konwencjonalny algorytm ewolucyjny operuje na rozwiązaniach reprezentowanych w postaci ciągów kodowych o stałej długości. Jednak w proponowanym podejściu pojedynczy stan jest zbiorem, a efektywne poszukiwanie rozwiązań suboptymalnych wymaga przeglądania zbiorów o różnych licznościach. Zastosowanie konwencjonalnego algorytmu wraz z opisywanymi wyżej operatorami rekombinacji uniemożliwiałoby przeglądanie rozwiązań F o licznościach $|F| > |F|_0$ oraz wykorzystanie w rozwiązaniach tych cech z podstawowego zbioru cech F_0, które nie zostały wylosowane podczas tworzenia populacji początkowej. Stąd algorytm 5.2.2 wyposażony jest w dodatkowy krok przeprowadzany w każdej iteracji (pokoleniu) algorytmu ewolucyjnego, krok polegający na lokalnej optymalizacji osobników w populacji P. Wykorzystywany tu algorytm może być w zasadzie dowolny, istotne jest jedynie to, aby umożliwiał zmianę rozmiaru (liczności) rozwiązań (osobników) z P i w przypadku średnim nie pogarszał ich jakości (przystosowania E). W eksperymentach obliczeniowych opisywanych w rozdziale 6 zastosowano algorytm SLS. Podobne podejście nazywane jest często w literaturze algorytmem ewolucyjnym z lokalną optymalizacją rozwiązań (lub "z lokalnym przeszukiwaniem", ang. *genetic local search*) i, choć motywowane raczej dążeniem do uzyskiwania lepszych rozwiązań niż specyfiką ich kodowania, wykazało swoją przydatność w eks-

perymentach porównawczych z konwencjonalnym algorytmem ewolucyjnym [Merz & Freisleben 1997], [Jaszkiewicz 1999].

Wprowadzenie fazy lokalnej optymalizacji osobników prowadzi niestety do znacznego wzrostu złożoności obliczeniowej algorytmu 5.2.2. Stąd w praktyce stosuje się tu pewne ograniczenia, na przykład przeprowadzając optymalizację tylko wybranych (np. najlepiej przystosowanych) osobników, lub wykonując pojedynczą iterację algorytmu SLS w miejsce pełnego przeszukiwania stromego.

5.2.5 Funkcja oceniająca

Omawiane podejście nie narzuca żadnych wymagań na funkcję E oceniającą stan (zbiór cech). Stanowi ona element "wymienny" proponowanego podejścia i zasadniczo można stosować dowolną z funkcji wymienionych w punkcie 4.4.4. W dalszej części pracy ograniczamy się jednak tylko do podejścia *wrapper*, z racji jego skuteczności. Algorytm indukcji stosowany w tym podejściu może być zasadniczo dowolny, przy czym z praktycznych względów przydatne są szczególnie algorytmy charakteryzujące się niską złożonością obliczeniową procesu uczenia i testowania.

Do obliczania wartości funkcji E w metodzie *wrapper* w zastosowaniu omawianym w rozdziale 6 początkowo stosowano n_{CV}-krotną walidację skrośną (CV), dla $n_{CV} \in \langle 4, 10 \rangle$, co jest częstym podejściem w selekcji cech [Kohavi & John 1997]. Jednak eksperymenty obliczeniowe pokazały, że tak obliczona wartość E jest zbyt optymisyczną estymatą zdolności predykcyjnej zbioru cech (trafność klasyfikowania $\eta(T)$ była z reguły niższa o kilka-kilkanaście procent od wartości E dla wyindukowanego zbioru cech. Przyczyną tego zjawiska jest fakt, że w walidacji skrośnej każda para zbiorów uczących L_{l_1} i L_{l_2}, $l_1, l_2 \in \langle 1, n_{CV} \rangle$ różni się jedynie

$$(1 - (n_{CV} - 2)/n_{CV}) \cdot 100\% \tag{5.4}$$

przykładami (w stosunku do liczności całego zbioru uczącego $|L|$), czyli np. 20% przykładów ze zbioru L dla $n_{CV} = 10$. Aby zmniejszyć wariancję trafności predykcyjnej w poszczególnych eksperymentach walidacji skrośnej stosuje się zazwyczaj $n_{CV} \geq 10$. Jednak im większa wartość n_{CV}, tym większy stopień pokrywania się poszczególnych podzbiorów uczących, a w przypadku granicznym, gdy $n_{CV} = |L|$, każda para z nich różni się tylko jednym przykładem. W konsekwencji klasyfikatory indukowane w poszczególnych iteracjach są do siebie podobne, a obliczana w ten sposób funkcja oceniająca oblicza *de facto* zdolność predykcyjną klasyfikatora bazującego *na określonym zbiorze przykładów uczących*, co stanowi swoistą formę przeuczenia i jest bardzo niekorzystne z punktu widzenia rozważanego zadania, tj. oceny zbiorów *cech*.

Jeżeli zatem *wrapper* ma oceniać zdolność predykcyjną zbioru cech, to wartość E musi być możliwie niezależna od zbioru uczącego. Można to uzyskać zapewniając większą rozłączność poszczególnych par podzbiorów uczących L_{l_1} i L_{l_2}. Jak pokazuje wyrażenie 5.4, można by to uzyskać zmiejszając wartość n_{CV}, to jednak zwiększa wariancję trafności klasyfikowania η_{CV}. Stąd w proponowanym podejściu stosowana jest metoda nazwana roboczo *odwrotną walidacją skrośną* (ang. *inverted cross-validation*, *ICV*), gdzie role podzbiorów uczących i podzbiorów testujących są ze sobą zamienione, tj. w l-tej iteracji ICV klasyfikator uczy się na podzbiorze T_l i jest testowany na podzbiorze L_l. W konsekwencji podzbiory używane do uczenia są parami rozłączne, a rozważany zbiór cech dla uzyskania wysokiej wartości E musi pozwolić na wyindukowanie z różnych zbiorów uczących n_{CV} niezależnych klasyfikatorów o dobrej zdolności predykcyjnej. Eksperymenty obliczeniowe pokazały, że tak obliczona wartość E jest bardzo dobrą estymatą zdolności predykcyjnej zbioru cech i nie odbiega od trafności klasyfikowania uzyskiwanej na niezależnym zbiorze testującym o więcej niż 1-2% (por. rozdział 6). Innym, niejako ubocznym następstwem użycia ICV w miejsce CV jest zmniejszenie czasochłonności obliczeń, ponieważ dla większości stosowanych klasyfikatorów złożność obliczeniowa procesu uczenia jest większa (w funkcji rozmiaru zbioru uczącego) od złożoności procesu testowania (por. punkt 5.4), a w ICV pojedyczny zbiór używany do uczenia jest $n_{CV} - 1$ krotnie mniejszy niż w przypadku CV[9].

5.3 Ukierunkowanie indukcyjne podejścia

Przez ukierunkowanie indukcyjne (UI) konwencjonalnego (w sensie: pozbawionego mechanizmu KI) algorytmu uczenia maszynowego rozumie się sposób wybierania (preferowania) hipotez z przestrzeni hipotez H (por. punkt 4.3). W systemach uczących się wykorzystujących KI interpretacja tego pojęcia jest nieco inna. W szczególności, ukierunkowanie indukcyjne omawianego podejścia jest związane bardziej z preferencjami systemu uczącego się dotyczącymi *wyboru przestrzeni reprezentacji* (zbioru cech) w F. W związku z tym być może bardziej stosowny byłby tu termin "ukierunkowanie reprezentacyjne" lub "ukierunkowanie konstruktywnej indukcji", ponieważ jednak ukierunkowanie indukcyjne algorytmu indukcji stosowanego w funkcji oceniającej E (*wrapper*) stanowi istotny element metody, zostaniemy przy określeniu "ukierunkowanie indukcyjne".

Ukierunkowanie indukcyjne opisywanego podejścia KI jest wypadkową ukierun-

[9]Warto też zaznaczyć, że zarówno w przypadku CV jak i ICV, dla zapewnienia porównywalności ocen różnych zbiorów cech mierzonych funkcją E, podział na podzbiory T_l musi pozostawać niezmieniony w procesie przeszukiwania.

kowań powiązanych z poszczególnymi jej elementami, tj.:

1. Podstawową reprezentacją obrazu $r(x)$,
2. Operatorami KI (zbiory F_0 i O),
3. Algorytmem przeszukiwania przestrzeni stanów,
4. Funkcją oceniającą E.

Ponieważ elementy wymienione w punktach 1) i 2) zależą od rozważanego zastosowania, nie da się analizować ich ukierunkowania w ogólnym przypadku, można to jednak zrobić dla konkretnego zastosowania. Na przykład dla rozważanego w rozdziale 6 problemu rozpoznawania ręcznie pisanych znaków ukierunkowanie reprezentacji podstawowej obrazu $r(x)$ przejawia się m.in. w tworzeniu składowych pierwotnych wyłącznie na podstawie punktów obrazu należących do szkieletu analizowanego obiektu.

Ukierunkowania powiązane z punktami 1) i 2) mają charakter *twardy*, ponieważ ograniczają przestrzeń rozważanych reprezentacji F. Z kolei w odniesieniu do pozostałych dwóch elementów można mówić o ukierunkowaniach *miękkich* (por. punkt 4.3).

Ad.3) Ukierunkowanie indukcyjne **algorytmu przeszukiwania stromego (SLS)** przejawia się w **preferowaniu mało licznych zbiorów cech (reprezentacji), złożonych z cech krótkich** (prostych). Wynika to z faktu, że przeszukiwanie rozpoczyna się od stanu $F_0 = \emptyset$ i ma charakter lokalny (por. algorytm 5.2.1). W konsekwencji dużo bardziej prawdopodobne jest wygenerowanie *prostej reprezentacji*, co stanowi korzystną analogię do preferowania *prostych hipotez* w konwencjonalnych technikach uczenia maszynowego. Jednak z drugiej strony brak możliwości przeprowadzania nawrotów implikuje podatność na problem lokalnych optimów.

Ukierunkowanie indukcyjne **algorytmu ewolucyjnego (AE)** ma bardziej złożony charakter i wiąże się ze znanym twierdzeniem o preferowaniu krótkich i "silnych" schematów (tzw. twierdzenie o schematach, por. np. [Michalewicz 1996], s. 80). Jednak w związku ze specyficzną reprezentacją rozwiązań w omawianym podejściu, gdzie nie można mówić o schematach (ciągach symboli $0, 1$ i $*$) w oryginalnym dla AE znaczeniu tego terminu, nie da się tej prawidłowości przenieść wprost. Algorytm 5.2.2 z definicji preferuje oczywiście rozwiązania (osobników) o wysokich wartościach funkcji przystosowania E. Jednocześnie należy się spodziewać, że faza lokalnej optymalizacji zapobiegać będzie nadmiernemu rozrostowi (w sensie liczby cech) rozwiązań w populacji P.

Ad.4) Ukierunkowanie związane z funkcją oceniającą E zależy od charakteru algorytmu indukcji użytego w metodzie wrapper. W przypadku użycia algorytmu

kNN wysoko oceniane będą reprezentacje (zbiory cech), w których przykłady należące do poszczególnych klas decyzyjnych tworzą zwarte *skupienia*, co wynika z tego, że metoda ta bazuje na wzajemnych odległościach przykładów w przestrzeni cech. Z kolei, gdy użyje się generatora drzew decyzyjnych, preferowane są reprezentacje, w których klasy decyzyjne da się od siebie odseparować przez definiowanie *progów* na poszczególnych atrybutach warunkowych, co wynika ze sposobu traktowania atrybutów ciągłych w tym podejściu.

5.4 Analiza złożoności obliczeniowej

Złożoność obliczeniowa proponowanej metody zależy przede wszystkim od algorytmu przeszukiwania przestrzeni rozwiązań (stanów) $\mathcal{F}$. Ponieważ oszacowanie złożoności obliczeniowej algorytmu ewolucyjnego jest w ogólności trudne, poniższa analiza ogranicza się jedynie do przypadku, w którym stosowanym algorytmem jest strome przeszukiwanie lokalne (SLS).

Poza algorytmem przeszukiwania pewien wpływ na czas trwania obliczeń mają **operacje przeprowadzane w ramach każdego ocenianego stanu** F_k, tj.

1. obliczanie wartości cech z F_k dla przykładów ze zbioru uczącego,
2. obliczanie wartości funkcji oceniającej $E(F_k)$.

Jeżeli przez n_N oznaczymy liczbę ocenianych stanów, czyli

$$n_N = \sum_{k=0}^{k^+} |N(F_k)|,$$

gdzie F_{k^+} jest końcowym stanem procesu przeszukiwania przestrzeni $\mathcal{F}$, a przez n_F i n_E odpowiednio liczbę elementarnych operacji algorytmu obliczania wartości cech i algorytmu obliczania wartości $E(F_k)$, wówczas całkowitą złożoność obliczeniową metody da się przedstawić jako

$$O(n_N(n_F + n_E)). \tag{5.5}$$

Zaprezentowana niżej analiza prowadzi do wyrażenia poszczególnych wielkości składających się na złożoność w kategoriach

- *rozmiaru* instancji problemu, tj. głównie wielkości zbioru przykładów X i użytych operatorów KI $O = A \cup S$, oraz

- *ograniczeń* narzuconych na proces przeszukiwania przestrzeni stanów, tj. głównie ograniczenia liczności konstruowanego zbioru cech $|F|_{\max}$ i maksymalnej długości cechy $|f|_{\max}$.

Zgodnie z uwagą poczynioną w punkcie 5.2.3, zakładamy dalej, że zbiór wykorzystywanych operatorów $O = S \cup A$ zawiera tylko operatory, dla których cecha atomowa a_i jest ustalona. Stąd zbiór cech atomowych A_0 nie pojawia się w poniższej analizie.

Ad.1) Obliczenie wartości cechy dla pojedynczego przykładu wymaga zastosowania co najwyżej $|f|_{\max}$ operatorów selekcji i agregacji, gdzie $|f|_{\max}$ jest górnym ograniczeniem na długość cechy. W zależności od rozważanego zbioru operatorów (dobranego np. do zastosowania), obliczenia przeprowadzane w ramach pojedynczego operatora mogą mieć różną złożoność; oznaczmy przez n_{sel} maksymalną liczbę elementarnych operacji wymaganych do zastosowania pojedynczego operatora selekcji, zaś przez n_{agr} analogiczną wielkość dla operatorów agregacji. Zatem algorytm obliczający wartość pojedynczej cechy wymaga $|f|_{\max} n_{sel} + n_{agr}$ operacji, a dla danego zbioru cech F jest to w najgorszym przypadku

$$n_F = |F|_{\max} (|f|_{\max} n_{sel} + n_{agr})$$

■

Ad.2) Proponowane podejście nie narzuca ograniczeń na charakter stosowanej funkcji oceniającej E. W eksperymentach opisywanych w rozdziale 6 stosowano podejście *wrapper*, które wymaga liczby operacji określonych następującym wyrażeniem:

$$n_E = n_{CV} (n_{train} + n_{test}), \tag{5.6}$$

gdzie n_{train} i n_{test} są odpowiednio liczbą operacji algorytmu uczenia i algorytmu testowania, a n_{CV} to liczba eksperymentów walidacji skrośnej. W zależności od zastosowanego algorytmu indukcji, wielkości te mogą się znacznie różnić, ale w ogólności, ze względu na wykorzystanie heurystyk w uczeniu i testowaniu klasyfikatorów, są to wyrażenia będące **wielomianowymi funkcjami** parametrów metody. W opisywanych eksperymentach stosowano generator drzew decyzyjnych i algorytm kNN, dla których wielkości te to (dla zbioru cech F, zbioru uczącego L i zbioru testującego T):

- drzewa decyzyjne:
  - $n_{train} = |F|\,|L| \log_2 |L| + |F| \log_2 |F|$ (liczba cech × koszt posortowania (algorytmem *quicksort*) wartości cechy w celu wyznaczenia punktu cięcia + liczba cech × maksymalna liczba węzłów drzewa binarnego, przy założeniu o najwyżej jednokrotnym wykorzystaniu każdej cechy),

  - $n_{test} = |T|\,|F|$ (rozmiar zbioru testującego $\times$ ilość testów przeprowadzanych na poszczególnych cechach, niezbędnych do zaklasyfikowania przykładu, czyli maksymalna długość ścieżki od korzenia do liścia w drzewie),

- algorytm kNN:
  - $n_{train} = |L|\,|F|$ (rozmiar opisu zbioru uczącego, zapamiętywanego w całości przez klasyfikator),
  - $n_{test} = |T|\,|L|\,|F|$ (rozmiar zbioru testującego $\times$ ilość zapamiętanych przykładów $\times$ koszt obliczenia odległości od klasyfikowanego przykładu),

Spośród wymienionych tu algorytmów największą złożonością obliczeniową charakteryzuje się proces indukcji drzewa decyzyjnego. W zamian za to drzewa oferują bardzo niski koszt obliczeniowy algorytmu klasyfikowania. W przypadku algorytmu kNN brak jest znaczącej różnicy w złożoności obliczeniowej procesu uczenia i testowania. Ponieważ zazwyczaj $|L| \gg |F|$, *wrapper* wykorzystujący drzewa decyzyjne jest jednak w praktyce szybszy. **Dla potrzeb dalszej analizy załóżmy wykorzystanie drzew decyzyjnych**. Wówczas, zgodnie z 5.6, liczba operacji wymaganych do obliczenia wartości funkcji E wynosi maksymalnie (tj. dla maksymalnie licznego zbioru ocenianych cech, tj. $|F|_{\max}$)

$$\begin{aligned} n_E &= n_{CV}\left(|F|_{\max}|L|\log_2|L| + |F|_{\max}\log_2|F|_{\max} + |T|\,|F|_{\max}\right) \\ &= n_{CV}\,|F|_{\max}\left(|L|\log_2|L| + \log_2|F|_{\max} + |T|\right) \end{aligned}$$

Bez utraty ogólności załóżmy, że $|T| = |X|/n_{CV}$ i $|L| = |X|\,(n_{CV}-1)/n_{CV}$, tj. że oceniany zbiór przykładów X da się podzielić na n_{CV} równo licznych podzbiorów w procesie walidacji skrośnej. n_E przyjmuje wówczas wartość

$$n_E = |F|_{\max}\left[|X|\,(n_{CV}-1)\log_2\frac{|X|\,(n_{CV}-1)}{n_{CV}} + n_{CV}\log_2|F|_{\max} + |X|\right]$$

■

W proponowanej metodzie zastosowano strome przeszukiwanie lokalne do przeglądania przestrzeni stanów. Przedstawiona tu analiza złożoności dotyczy najprostszej wersji tego algorytmu (SLS), która nie implementuje nawrotów i zatrzymuje się, gdy z bieżącego stanu nie da się wykonać żadnego ruchu polepszającego (zwiększającego) wartość funkcji oceniającej E. Przy oszacowaniu liczby stanów przeglądanych przez SLS warto posłużyć się spostrzeżeniem, iż stosowana funkcja oceniająca *wrapper*, będąc wynikiem eksperymentu walidacji skrośnej, ma w rzeczywistości charakter dyskretny (por. wyrażenie 4.3):

$$E(F) = \eta_{CV}(h_F, X) \in \left\{\frac{l}{|X|}, l = 0, 1, 2, \ldots, |X|\right\}$$

W konsekwencji, przy założeniu że $E(F_0) = 0$, **algorytm przeszukiwania przestrzeni stanów może wykonać co najwyżej** $|X|$ **ruchów**; ma to miejsce w skrajnym przypadku, gdy przy wykonywaniu każdego ruchu wartość E wzrasta o najmniejszą możliwą wielkośc, tj. $1/|X|$.

W każdym z odwiedzanych stanów F_k przeglądane jest jego sąsiedztwo $N(F_k)$ (patrz 5.3). Rozmiar sąsiedztwa zależy od zbioru dopuszczalnych ruchów i zmienia się w zależności od stanu, co przysparza pewnych kłopotów w ocenie złożoności obliczeniowej i implikuje konieczność używania górnych ograniczeń (bardzo zawyżonych w porównaniu z przypadkiem średnim). W szczególności, górne ograniczenie na liczbę ruchów dotyczących **całych cech** to $|F_0| + |F|_{\max}$ (dodawanie do bieżącego rozwiązania cech ze zbioru F_0 i usuwanie cech z bieżącego rozwiązania, którego liczność to co najwyżej $|F|_{\max}$). Z kolei górne ograniczenie na liczbę ruchów polegających na modyfikacji **pojedynczej** cechy to

- $|O|\,|f|_{\max}$ dla dodawania (wstawiania) operatorów,
- $|O|\,|f|_{\max}$ dla zastępowania operatorów[10],
- $|f|_{\max}$ dla usuwania operatorów.

Razem daje to zatem $2\,|O|\,|f|_{\max} + |f|_{\max}$ możliwych ruchów polegających na modyfikacji pojedynczej cechy. Cech w bieżącym rozwiązaniu może być natomiast co najwyżej $|F|_{\max}$. Zatem **górne ograniczenie na liczbę stanów ocenianych** podczas całego procesu przeszukiwania przestrzeni (lub inaczej: górne ograniczenie na liczbę wywołań funkcji E) to

$$\begin{aligned} n_N &= |X|\,[|F_0| + |F|_{\max} + |F|_{\max}\,(2\,|O|\,|f|_{\max} + |f|_{\max})] \\ &= |X|\,[|F_0| + |F|_{\max}\,(2\,|O|\,|f|_{\max} + |f|_{\max} + 1)] \end{aligned} \tag{5.7}$$

Całkowitą liczbę operacji algorytmu można zatem wyrazić jako (por wzór 5.5)

$$\begin{aligned} n_N\,(n_F + n_E) &= |X|\,[\,|F_0| + |F|_{\max}\,(2\,|O|\,|f|_{\max} + |f|_{\max} + 1)\,] \\ &\quad \cdot[\,|F|_{\max}\,(|f|_{\max}\,n_{sel} + n_{agr}) \\ &\quad + |F|_{\max}\,(|X|\,(n_{CV} - 1)\log_2 \frac{|X|\,(n_{CV} - 1)}{n_{CV}} + n_{CV}\log_2 |F|_{\max} + |X|)] \end{aligned}$$

co pozwala na ujęcie złożoności obliczeniowej podejścia w następujący sposób (dla przeszukiwania SLS i funkcji oceniającej typu *wrapper* z użyciem algorytmu

[10] Z pewnym przybliżeniem, nie uwzględniono tu bowiem osobno selektorów i agregatorów.

indukcji drzew decyzyjnych, przy założeniu że złożoność obliczeń przeprowadzanych w procesie selekcji pola widzenia i agregacji jest zaniedbywalna):

$$\begin{aligned} & O\left(n_N\left(n_F+n_E\right)\right) \\ = \; & O[\,|X|\,|F|_{\max}\left(|F_0|+|F|_{\max}\,|O|\,|f|_{\max}\right) \\ & \cdot\left(|f|_{\max}\,n_{sel}+n_{agr}+n_{CV}\left(|X|\log_2|X|+\log_2|F|_{\max}\right)\right)\,] \end{aligned} \tag{5.8}$$

Podsumowując, ze wzoru 5.8 wynika, iż czas działania algorytmu zależy:

- **liniowo** (x) od liczby stosowanych operatorów $|O|$, rozmiaru zdefiniowanego przez użytkownika zbioru cech podstawowych F_0 oraz liczby eksperymentów walidacji skrośnej n_{CV},
- **w stopniu** x^2 od maksymalnej długości cechy $|f|_{\max}$,
- **w stopniu** $x^2 \log_2 x$ od maksymalnej liczby cech $|F|_{\max}$ oraz liczby przykładów (rozmiaru zbioru uczącego) $|X|$.

Jest to zatem złożoność **wielomianowa**, akceptowalna z obliczeniowego punktu widzenia. Ponadto należy pamiętać, iż oszacowanie liczby ocenianych stanów n_N, która ma krytyczne znaczenie dla złożoności algorytmu, jest skrajnie pesymistyczne. Na przykład dla $|X| = 500$, $|F_0| = 20$, $|O| = 40$, $|F|_{\max} = 6$, $|f|_{\max} = 4$, liczba ocenianych stanów $n_N \equiv 1.1 \times 10^6$, podczas gdy eksperymenty obliczeniowe opisywane w rozdziale 6 pokazują, że przy takich parametrach problemu liczba ocenianych stanów nie przekracza kilkudziesięciu tysięcy.

Oszacowaną liczbę stanów rozważanych w procesie KI warto - choćby z pobudek poznawczych - **porównać z rozmiarem przestrzeni wszystkich stanów** (zbiorów cech) przy zadanych ograniczeniach. Liczba wszystkich możliwych cech o długości co najwyżej $|f|_{\max}$, zbudowanych z selektorów ze zbioru S i agregatorów ze zbioru A to

$$q_f\left(|f|_{\max}, A, S\right) = \sum_{l=0}^{|f|_{\max}-1} |S|^l\,|A| = \left(|S|^{|f|_{\max}}-1\right)|A|\,.$$

Zatem liczba wszystkich niepustych podzbiorów takich cech o liczności co najwyżej $|F|_{\max}$ to

$$q\left(|F|_{\max}, |f|_{\max}, A, S\right) = \sum_{l=1}^{|F|_{\max}} \binom{q_f\left(|f|_{\max}, A, S\right)}{l}$$

co już dla małych zbiorów A, S i silnych ograniczeń jest olbrzymią wielkością. Kontynuując podany wyżej przykład, jeżeli założyć że wśród $|O| = 40$ operatorów KI jest $|S| = 20$ selektorów i $|A| = 20$ agregatorów, $q_f\left(|f|_{\max}, A, S\right) = 3\,199\,980$, a $q\left(|F|_{\max}, |f|_{\max}, A, S\right) \equiv 1.49 \times 10^{36}$. Porównanie tej wartości z oszacowaną liczbą n_N stanów przeglądanych przez proponowany algorytm prowadzi do konkluzji, że algorytm przegląda bardzo małą część przestrzeni rozwiązań dopuszczalnych (dla powyższego przykładu ok. 10^{-30}), która w ogólności jest bardzo liczna. Widać zatem, że z praktycznego punktu widzenia zastosowanie podejścia heurystycznego ma tu na celu nie tyle przyspieszenie obliczeń, co ich umożliwienie, ponieważ zastosowanie pełnego przeszukiwania jest nierealne już dla bardzo małych instancji problemu.

Rozdział 6

Przykład zastosowania proponowanego podejścia w rozpoznawaniu ręcznie pisanych znaków alfanumerycznych

Niniejszy rozdział opisuje w szczegółach zastosowanie proponowanego podejścia w rozpoznawaniu ręcznie pisanych znaków alfanumerycznych (ang. *handwritten character recognition, HCR*). Rozdział zawiera krótkie wprowadzenie w zagadnienie HCR, a następnie opisuje w jaki sposób proponowana metoda została dostosowana do tego problemu. Dalej następuje opis analizowanego zbioru przykładów i sposobu przeprowadzenia obliczeń oraz prezentacja i dyskusja wyników eksperymentalnych.

6.1 Rozpoznawanie ręcznie pisanych znaków alfanumerycznych

Rozpoznawanie ręcznie pisanych znaków alfanumerycznych należy do najbardziej popularnych zastosowań analizy i rozpoznawania obrazów. Przyczyniają się do tego następujące czynniki:

- liczne możliwości zastosowań (rozpoznawanie kodów pocztowych, automatyczna analiza formularzy, autoryzacja na podstawie podpisu, itp.),
- niewielki rozmiar analizowanych obrazów (zazwyczaj nie więcej niż 100×100 punktów obrazu) i ich dwuwymiarowa charakterystyka (w odróżnieniu od za-

stosowań, gdzie obraz jest pewną projekcją rzeczywistości trójwymiarowej na dwa wymiary przestrzenne, np. robotyki),

- względna łatwość pozyskania danych (obrazów).

W literaturze spotkać można opisy wielu metod dedykowanych do tego zastosowania [Kato, Omachi, *et al.* 1999], [Cai & Liu 1999], [Wong & Chan 1998], przy czym szczególnie popularne są podejścia z użyciem sztucznych sieci neuronowych [Burr 1988], [LeCun & et al. 1989], [Jackel & et al. 1995], [LeCun & et al. 1995], [Żurada, Barski, *et al.* 1996] (rozdział 8) oraz podejścia strukturalne (np. [Zabawa 1994]). Szeroko przyjął się m.in. podział metod ze względu na:

- charakter zadania:
  - rozpoznawanie pisma ciągłego (całych słów zdań, itd.),
  - rozpoznawanie znaków izolowanych (np. w automatycznym wczytywaniu formularzy),
- charakter danych wejściowych:
  - rozpoznawanie na podstawie obrazu statycznego (mapy bitowej) - ang. *off-line HCR*,
  - rozpoznawanie na podstawie zapisu przebiegu pióra - ang. *on-line HCR* (wymaga co prawda specjalnego urządzenia wejściowego (ang. *tablet*), ale dostarcza za to dane dotyczące "trajektorii" pióra, jego prędkości i siły nacisku, itp.).

Warto nadmienić, iż HCR jest z punktu widzenia uczenia maszynowego zastosowaniem bardzo wymagajacym jeżeli chodzi o trafność klasyfikowania, ponieważ w praktyce (np. w tekście) rozpoznaje się zazwyczaj jednocześnie wiele znaków. Tam, gdzie rozpoznaje się słowa lub nawet całe zdania, część błędów popełnianych przez system da się wykryć (i często skorygować) przez stosowanie dodatkowych ograniczeń i baz wiedzy (np. słowników, reguł gramatycznych, etc.). Wciąż jednak pozostaje wiele zastosowań, gdzie brak jest dodatkowych możliwości korekcji błędu (np. przy rozpoznawaniu/wczytywaniu danych liczbowych) i w związku z tym prawidłowe rozpoznawanie znaków jest krytyczne. Opisywane zastosowanie odpowiada właśnie takiej sytuacji, dotyczy bowiem rozpoznawania *off-line* cyfr izolowanych.

6.2 Opis eksperymentów

6.2.1 Dane

W ramach niniejszej pracy przetestowano opisywane podejście na problemie rozpoznawania *off-line* znaków izolowanych. Analizowany zbiór przykładów to baza danych MNIST udostępniona w Internecie przez AT&T Labs-Research [LeCun & et al. 1995] (http://www.research.att.com/~yann/ocr/mnist/). Zawiera ona w sumie 70.000 obrazów cyfr od 0 do 9 i stanowi kompilację dwóch zbiorów danych: 35.000 przykładów z tzw. bazy SD-1 (*Special Database*) i 35.000 przykładów z bazy SD-3, udostępnianych przez amerykański Narodowy Instytut Standardyzacji (*National Institute for Standard Technology, NIST*). Obrazy obecne w bazie SD-1 zostały pozyskane wśród studentów uczelni wyższych, natomiast SD-3 wśród pracowników amerykańskiego odpowiednika Urzędu Statstycznego (*Census Bureau*). W sumie baza zawiera przykłady cyfr pisanych przez około 250 osób. Baza podzielona jest a priori na zbiór uczący L (60.000 przykładów) i testujący T (10.000 przykładów), z zachowaniem proporcji SD-1 i SD-3. Należy szczególnie podkreślić, iż zbiory osób "reprezentowanych" w zbiorze uczącym i testującym są rozłączne (cyfry zapisane przez daną osobę mogą się znaleźć tylko w jednym ze zbiorów: uczącym lub testującym).

Oryginalne obrazy w bazie NIST zostały pozyskane przez skanowanie dokumentów z rozdzielczością 300 punktów obrazu na cal (dpi). Każdy obraz z oryginalnej bazy NIST został wstępnie przetworzony przez twórców bazy MNIST. Przetwarzanie to polegało na normalizacji rozmiaru znaku tak, aby mieścił się on w obrazie o rozmiarach 20×20 punktów (przy zachowaniu proporcji w osi X i Y, czyli tzw. *aspect ratio*). Dla zachowania jak największej ilości informacji przy normalizacji stosuje się wygładzanie obrazu (ang. *antialiasing*), w wyniku czego obraz wynikowy zawiera 256 stopni szarości, choć oryginalne obrazy są binarne (czarno-białe). Następnie obliczany jest środek masy (średnia ważona punktów obrazu) tak powstałego obrazu. Dalej obraz jest tak pozycjonowany, aby jego środek masy przypadał w centrum obrazu o rozmiarach 28×28 punktów. Ostatecznie baza danych MNIST zawiera zatem ustandaryzowane obrazy z gradacją 256 stopni szarości o rozmiarach 28×28 punktów każdy.

W bazie MNIST występuje dziesięć w przybliżeniu równolicznych klas decyzyjnych, każda dla jednej cyfry od 0 do 9. Przynależność przykładów do klas decyzyjnych została sprawdzona ręcznie przez ekspertów. Rys. 6.1 prezentuje pierwsze 100 obrazów ze zbioru uczącego.

Należy zaznaczyć, iż MNIST nie zawiera jedynie przykładów cyfr napisanych wyraźnie, tj. w sposób nie sprawiający trudności w rozpoznaniu przez człowieka.

Rysunek 6.1: Przykładowe obrazy z bazy danych MNIST

Rysunek 6.2 prezentuje wybrane "trudne" przykłady, których jest zdecydowanie więcej w części bazy MNIST utworzonej z bazy SD-1, czyli znaków pisanych przez studentów.

6.2.2 Sposób zastosowania proponowanego podejścia

Wstępne przetwarzanie obrazu

Ponieważ w rozpoznawaniu znaków alfanumerycznych istotny jest przede wszystkim kształt, proces wstępnego przetwarzania obrazu prowadzący do utworzenia podstawowej reprezentacji obrazu powinien zachowywać przede wszystkim ten aspekt obrazu. W obrazach znaków alfanumerycznych linie składające się na kształt znaku są zazwyczaj zdecydowanie grubsze niż jeden punkt obrazu (dla rozważanej bazy danych jest to najczęściej kilka punktów). Dlatego wstępne przetwarzanie rozpoczyna się od zastosowania algorytmu szkieletonizacji (osi środkowej, ang. *skeletonization, medial axis*), która polega na wyróżnieniu tzw. "szkieletu" obiektu. Do szkieletu należy każdy punkt p obszaru, dla którego istnieją dwa lub więcej punkty na brzegu obszaru, których odległość od p jest minimalna[1]. Do szkieletonizacji użyty został al-

[1]Znaną i trafną ilustracją tej operacji jest analogia "wypalania trawy": jeżeli analizowany kształt interpretować jako wysuszony trawnik podpalany w tej samej chwili we wszystkich punktach brzegu, to punkty, w których spotyka się ogień, wyznaczają szkielet kształtu.

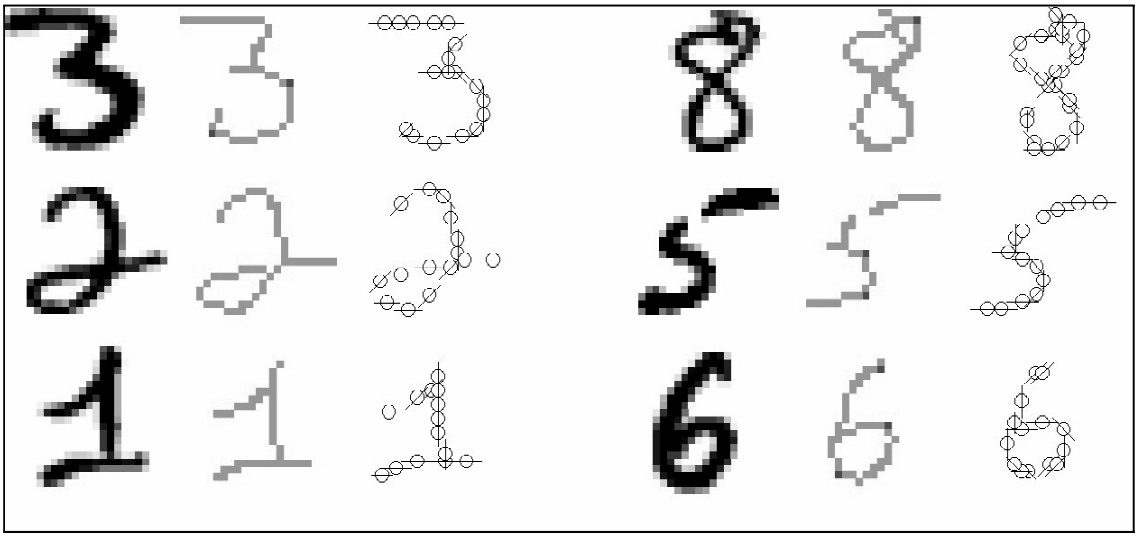
Rysunek 6.2: Wybrane trudne przykłady z bazy MNIST

gorytm znany z literatury (np. [Gonzalez & Woods 1992]), którego działanie polega na kontrolowanej erozji obszaru. Ponieważ jest on jednak przeznaczony do obliczania szkieletu w obrazach binarnych (dwuwartościowych), dla rozważanych obrazów ze stopniami szarości został tak zmodyfikowany, aby usuwać punkty w porządku odpowiadającym stopniu ich przynależności do figury, tj. od najjaśniejszych (najniższy stopień przynależności) do najciemniejszych (najwyższy stopień przynależności).

Podstawowa reprezentacja obrazu

W opisywanym zastosowaniu podstawowa reprezentacja $r(x)$ obrazu (por. punkt 5.2.1) jest *zbiorem* składowych pierwotnych i budowana jest na podstawie szkieletu obrazu (patrz poprzedni punkt). Dla każdego z wybranych ze szkieletu (według pewnej heurystycznej procedury) punktów p tworzona jest składowa pierwotna $a(p)$ reprezentująca punkt p oraz jego lokalne otoczenie, którego rozmiar w eksperymentach opisywanych w dalszej części pracy został ustalony na 3×3 punkty obrazu. Ponieważ jedną z istotniejszych informacji opisujących otoczenie punktu p jest lokalny kierunek szkieletu, pojedyncza składowa pierwotna $a(p)$ odpowiada w przybliżeniu spotykanemu często w literaturze pojęciu *linelet* (por. np. [Zhu 1999]).

Rysunek 6.3 prezentuje m.in. obrazy wybranych cyfr wraz z ich szkieletami. Po przeprowadzeniu szkieletonizacji eliminowane są jeszcze punkty, które posiadają dokładnie dwóch sąsiadów w 4-sąsiedztwie w kierunkach różniących się o 90° (czyli w taki sposób, aby każdy punkt miał co najwyżej dwóch sąsiedów w tzw. sąsiedztwie mieszanym, por. [Gonzalez & Woods 1992]). Na rysunku punkty te oznaczone są nieco ciemniejszym kolorem. Operacja ta ma na celu zredukowanie liczby składo-



Rysunek 6.3: Wynik wstępnego przetwarzania (szkielet) i podstawowa reprezentacja obrazu dla wybranych obrazów

wych pierwotnych. Ponadto dla każdego obrazu x, na rysunku pokazano graficzną interpretację jego podstawowej reprezentacji $r(x)$. Każda składowa pierwotna ilustrowana jest okręgiem, zaś dominujący kierunek - odcinkiem o odpowiednim nachyleniu.

W rozważanym zastosowaniu zdecydowano się zdefiniować następujace cechy atomowe składowej pierwotnej a (w nawiasach prostokątnych podano symbole stosowane w implementacji komputerowej, patrz dodatek A):

- współrzędna pozioma [x],
- współrzędna pionowa [y],
- kierunek (orientacja) [d],
- inne, np. szerokoś, wysokoś, współrzędne względem bieżącego pola widzenia [X,Y], pole powierzchni minimalnego prostokąta obejmującego (ang. *minimal bounding rectangle*) [A].

Dla uproszczenia rozważano tylko cztery wartości kierunku składowej pierwotnej, odpowiadające nachyleniom -90°, -45°, 0°, 45° względem dodatniego zwrotu osi poziomej. Dominujący kierunek w otoczeniu wyznaczany jest przy pomocy transformaty Hougha [Hough 1962], [Duda & Hart 1972] i następnie zaokrąglany do jednej

z czterech wymienionych wyżej wartości. Zatem zbiór cech atomowych wykorzystywanych w rozważanym zastosowaniu można zapisać z użyciem notacji stosowanej w implementacji komputerowej jako $A_0 = \{x, y, d, eX, eY, X, Y, A\}$. Dla ujednolicenia cech atomowych poddane są one normalizacji, zatem $\alpha(a) \in \langle 0, 1\rangle \, \forall \alpha \in A_0$, $a \in r(x)$, $\forall x \in X$.

Operatory agregacji

W rozważanym zastosowaniu wykorzystano następujące operatory agregacji wartości cech atomowych $\alpha \in A_0$ składowych pierwotnych a (w nawiasach prostokątnych podano symbole stosowane w implementacji komputerowej, patrz dodatek A):

- wartość maksymalna cechy atomowej α: $max_\alpha[R(x)] = \max_{a \in R(x)} \alpha(a)$ [M],
- wartość minimalna cechy atomowej α: $min_\alpha[R(x)] = \min_{a \in R(x)} \alpha(a)$ [m],
- wartość średnia cechy atomowej α: $avg_\alpha[R(x)] = \frac{1}{|R(x)|} \sum_{a \in R(x)} \alpha(a)$ [a],
- odchylenie standardowe wartości cechy atomowej α: dev_α [d],
- ilość składowych pierwotnych: $num[R(x)] = |R(x)|$ [n],

gdzie $R(x)$ jest polem widzenia. Zgodnie z uwagą zamieszczoną w punkcie 5.2.3, za wyjątkiem ostatniego operatora *num* są to agregatory uniwersalne, które stają się konkretnymi operatorami w zbiorze O przez ustalenie ("podstawienie") cechy atomowej α. Zaznaczmy, że w praktyce przy obliczaniu wartości agregatora należy liczyć się z koniecznością obsługi pewnych sytuacji nietypowych, na przykład gdy pole widzenia jest puste. Ponieważ jest to jednak kwestia raczej techniczna, zostanie ona omówiona w jednym z punktów dodatku B.

Operatory selekcji pola widzenia

Operatory selekcji pola widzenia wykorzystane w opisywanym zastosowaniu uwzględniają *porządkowy* charakter wszystkich cech atomowych. Można je podzielić na dwie grupy:

1. Selektory *jednowymiarowe*, tj. dokonujące selekcji pola widzenia na podstawie jednej cechy atomowej,
2. Selektory *dwuwymiarowe*, tj. dokonujące selekcji pola widzenia na podstawie dwóch cech atomowych.

Selektory reprezentujące pierwszą grupę wybierają z bieżącego pola widzenia $R(x)$ te składowe pierwotne, dla których wartość pewnej cechy atomowej $\alpha \in A_0$ jest mniejsza (lub większa) od pewnego progu $\lambda \in \Re$. W związku z tym każdy z nich występuje w dwóch wersjach, z relacją "$\leq$" lub ">". Ogólną definicję rodziny operatorów selekcji jednowymiarowej $s_\alpha^{\pi,\lambda}$, $\pi \in \{\leq, >\}$, dokonujących wyboru składowych pierwotnych na podstawie porównywania ich cechy atomowej $\alpha \in A_0$ z wartością λ podaje wzór 6.1.

$$s_\alpha^{\pi,\lambda}[R(x)] = \{a \in R(x) : \alpha(a)\ \pi\ \lambda\} \tag{6.1}$$

Poszczególne operatory różnią się przede wszystkim sposobem ustalania wartości λ. Można je podzielić na takie, w których $\lambda = const$ (stosowane wartości stałe wymieniono w dodatku A), oraz takie, gdzie λ jest wartością obliczoną na podstawie wartości cechy atomowej α składowych pierwotnych $a \in R(x)$, np.:

- $\lambda = max_\alpha[R(x)]$,
- $\lambda = min_\alpha[R(x)]$,
- $\lambda = avg_\alpha[R(x)]$.

Selektory dwuwymiarowe reprezentują operatory dokonujące selekcji pola widzenia na podstawie współrzędnych przestrzennych, czyli cech atomowych oznaczonych przez x i y w notacji stosowanej w implementacji komputerowej. Selektory tego typu, z racji podobieństwa ich charakterystyki do tzw. pól recepcyjnych stosowanych w niektórych typach sztucznych sieci neuronowych (por. rozdział 3), nazywane będą dalej *selektorami pól recepcyjnych*. W eksperymentach zastosowano selektory o polach recepcyjnych o symetrii kołowej, których działanie charakteryzuje wzór 6.2.

$$s_{rec}^{x_0,y_0,\rho}[R(x)] = \left\{a \in R(x) : d^2(x_0, y_0, x_c(a), y_c(a)) \leq \rho\right\} \tag{6.2}$$

gdzie $\rho \in \langle 0, 1\rangle$ jest ustalonym *promieniem* pola recepcyjnego (stosowane wartości ρ wymieniono w dodatku A), (x_0, y_0) jest punktem centralnym pola recepcyjnego, $(x(a), y(a))$ to współrzędne kartezjańskie składowej pierwotnej a, a $d^2(x_0, y_0, x_1, y_1)$ jest odległością euklidesową pomiędzy punktami (x_0, y_0) i (x_1, y_1). W eksperymentach stosowano dwa selektory tego typu, różniące się sposobem ustalania punktu centralnego pola recepcyjnego (x_0, y_0) (w nawiasach prostokątnych podano oznaczenia stosowane w implementacji komputerowej, patrz dodatek A):

- selektor *bezwzględnego* pola recepcyjnego [R],
- selektor *względnego* pola recepcyjnego [r].

Selektor bezwzględnego pola recepcyjnego interpretuje wartości x_0 i y_0 w sposób bezwzględny, tj. jako współrzędne względem początku układu współrzędnych, w którym określona jest podstawowa reprezentacja obrazu. Natomiast selektor względny definiuje nowy układ współrzędnych, oparty na minimalnym prostokącie zawierającym składowe pierwotne w bieżącym polu widzenia i interpretuje wartości x_0 i y_0 jako (znormalizowane) współrzędne kartezjańskie w tym układzie. Wprowadzenie względnego pola recepcyjnego daje możliwość automatycznego dostosowywania pola widzenia do fragmentu rozpoznawanego obrazu (obiektu) i zapewnia w ten sposób pewną odporność metody na zniekształcenia wzorców.

6.2.3 Metodyka przeprowadzania eksperymentu

Z powodu ograniczeń dostępnej mocy obliczeniowej i wielkości pamięci operacyjnej, w eksperymentach nie wykorzystywano wszystkich 60.000 obrazów składających się na zbiór uczący. W miejsce tego stworzono zbiór uczący L zawierający 10.000 przykładów, po 1.000 pierwszych przykładów z oryginalnego zbioru uczącego dla każdej klasy decyzyjnej (są to zatem przykłady z bazy danych SD-3, tj. cyfr "trudniejszych", por. punkt 6.2.1). Natomiast do testowania, wobec dużo mniejszych wymagań na zasoby obliczeniowe, wykorzystywano pełen zbiór testujący T bazy danych MNIST, zawierający 10.000 przykładów.

Na każdy z opisywanych dalej eksperymentów składały się następujące **fazy**:

1. Konstruktywna indukcja cech obrazu na podstawie obrazów ze zbioru uczącego L, prowadząca do wyindukowania zbioru cech (reprezentacji) F^+,
2. Generowanie (uczenie) klasyfikatora h na zbiorze uczącym L z użyciem wyindukowanego zbioru cech F^+,
3. Testowanie otrzymanego klasyfikatora h na zbiorze testującym T.

Ze względu na znaczną czasochłonność obliczeń nie zastosowano metody walidacji skrośnej, która dzięki wielokrotnemu powtarzaniu cyklu uczenie-testowanie gwarantuje mniejszą wariancję wynków. W zamian za to jednak test jednokrotny daje możliwość porównania wyników z cytowanymi w [LeCun & et al. 1995] (patrz punkt 6.4). Ponadto należy się spodziewać, iż w związku ze znacznymi rozmiarami zbioru uczącego i testującego wyniki testu walidacji skrośnej nie odbiegałyby znacząco od prezentowanych poniżej.

Wyniki eksperymentów oceniano głównie wg następujących **kryteriów**:

- trafności klasyfikowania na zbiorze testującym T (zdolności predykcyjnej) wybranych klasyfikatorów uczonych na zbiorze uczącym L z użyciem reprezentacji F^+ ,

- zwięzłości opisu (liczba wyindukowanych cech $|F^+|$, rozkład długości $|f_j|$ cech $f_j \in F^+$),

Ponadto w niektórych eksperymentach brano także pod uwagę liczbę rozwiązań ocenianych w trakcie przeszukiwania przestrzeni rozwiązań.

Ad. 1) Dla zapewnienia porównywalności wyników, część parametrów pozostawała niezmienna podczas przeprowadzania eksperymentów. Należały do nich:

- liczność pojedynczej klasy decyzyjnej dla procesu KI: $|C_i| = 1000$,
- maksymalna liczba cech (rozmiar reprezentacji) $|F|_{\max} = 20$,
- maksymalna długość cechy $|f|_{\max} = 8$,
- liczba powtórzeń procesów uczenia i testowania dla walidacji skrośnej w podejściu *wrapper* $n_{CV} = 4$.

Wartość parametru n_{CV} została ustalona na podstawie wstępnych eksperymentów, które wykazały, że czterokrotna walidacja skrośna daje wartość E o wystarczająco niskiej wariancji.

Ze względu na liczbę parametrów metody i znaczną czasochłonność obliczeń, przeprowadzenie kompletnej serii eksperymentów dla stopniowo zmieniających się ich ustawień (tzw. *grid search*) byłoby niewykonalne. Przeprowadzono zatem serie eksperymentów mających na celu ustalenie wpływu pojedynczych parametrów metody na uzyskiwane wyniki. Najistotniejszymi z analizowanych parametrów (opcji) metody były:

- algorytm indukcji stosowany w funkcjach oceniających E (E_{kNN} i E_{DT}, punkt 6.3.1),
- algorytm przeszukiwania przestrzeni F (SLS i AE, punkt 6.3.2),
- operatory konstruktywnej indukcji cech.

Jako algorytm indukcji (*wrapper*) w funkcji oceniającej E wykorzystywano:

- metodę k najbliższych sąsiadów (ang. *k nearest neighbours*, *kNN*) ($k = 1$), jako podstawowego reprezentanta metod minimalnogodległościowych (por. np. rozdział 4 w [Tadeusiewicz & Flasiński 1991]),

- algorytm indukcji drzew decyzyjnych (DT) (zaimplementowano algorytm wykorzystujący zysk na entropii (ang. *gain ratio*) do oceniania potencjalnych podziałów węzłów, z upraszczaniem drzewa w trakcie jego budowania (ang. *pre-prunning*) opartym na kryterium *error-complexity* [Breiman, Friedman, *et al.* 1984]).

Funkcje oceniające bazujące na poszczególnych induktorach oznaczane będą dalej odpowiednio E_{kNN} i E_{DT}. Wybór tych algorytmów indukcji pokierowany był ich komplementarnością. Za metodą kNN przemawiały praktycznie jednostkowa złożoność obliczeniowa algorytmu uczenia (zapamiętanie zbioru uczącego L) oraz brak możliwości różnicowania ważności poszczególnych atrybutów, co m.in. zapewnia pożądany spadek wartości E_{kNN} w wypadku dodania do zbioru cech cechy zaszumionej. Wadą tego algorytmu indukcji jest stosunkowo duża złożoność procesu klasyfikowania (porównanie z wszystkimi przykładami ze zbioru uczącego). Z kolei zaletami algorytmu DT są niskie złożoności obliczeniowe zarówno procesu indukcji, jak i testowania. Jednocześnie jednak, z punktu widzenia podejścia *wrapper,* wadą tego induktora jest zdolność do selekcji cech (np. ignorowanie w budowie drzewa zaszumionych atrybutów), co uodpornia go na cechy nadmiarowe i zaszumione, prowadząc do potencjalnych zafałszowań wartości funkcji E.

Podstawowym parametrem metody jest zbiór cech podstawowych F_0 oraz zbiór operatorów KI O (por. punkt 5.2.4). Zgodnie z uwagą zawartą w punkcie 5.2.3, ze względów praktycznych zbiory te nie są definiowane explicite, lecz poprzez "szablony" (dodatek A), stanowiące zwięzłe, zbliżone do gramatyk formalnych, opisy zbiorów F_0 lub O. Manewrując szablonami można wpływać na zawartość i liczność sąsiedztwa N (wzór 5.3) przeglądanego w każdym stanie w algorytmie SLS lub na proces rekombinacji rozwiązań w algorytmie ewolucyjnym. W eksperymentach zastosowano trzy zestawy operatorów. Najprostszy, oznaczany dalej przez Z_A, charakteryzuje się niewielką licznością zbiorów F_0 i O (odpowiednio 20 i 129). Zestaw Z_B różni się od Z_A tym, że jego zbiór F_0 zawiera cechy poszerzone w stosunku do Z_A o operatory ze zbioru O zestawu Z_A (w konsekwencji $|F_0| = 2600$). Z kolei Z_C bazuje na dokładnie tych samych cechach F_0 i operatorach O co zestaw Z_A, ma jednak poszerzony w stosunku do niego zestaw cech atomowych A_0 (tutaj $|F_0| = 45, |O| = 257$). Pełną informację o zawartościach zbiorów F_0 i O znaleźć można w dodatku A.

Ad. 2) Do końcowej weryfikacji skuteczności (trafności klasyfikowania) wyindukowanych reprezentacji F^+ przeprowadzano eksperymenty uczenia na zbiorze L i testowania na zbiorze T wybranych algorytmów indukcji reprezentujących różne nurty charakterystyczne dla uczenia maszynowego. W tym celu wykorzystano algorytmy:

- algorytm k najbliższych sąsiadów kNN, $k = 1$,
- algorytm generowania drzew decyzyjnych C4.5 [Quinlan 1992] (wykorzystywano drzewo uproszczone wygenerowane przy domyślnych ustawieniach parametrów, tj. kryterium podziału węzła *gain ratio*, poziom ufności procedury upraszczającej drzewo 0.25),
- nieliniową warstwową sieć neuronową (SSN) uczoną algorytmem *resilient backpropagation* (*RPROP,* [Riedmiller & Braun 1992]), udoskonaloną wersja algorytmu wstecznej propagacji błędu.

6.3 Wyniki eksperymentów

Niniejszy podrozdział prezentuje wyniki **wybranych** eksperymentów obliczeniowych. Rezultaty i zagadnienia prezentowane w poszczególnych punktach odzwierciedlają ważniejsze etapy rozwoju, jaki przeszło proponowane podejście w ramach prac nad niniejszą rozprawą.

6.3.1 Porównanie funkcji oceniających wykorzystujących różne klasyfikatory w metodzie *wrapper* (E_{kNN} i E_{DT})

Tabele 6.1 i 6.2 prezentują wyniki wybranych eksperymentów KI z zastosowaniem SLS (punkt 5.2.4) jako algorytmu przeszukiwania przestrzeni rozwiązań oraz obu funkcji oceniających rozważanych w eksperymentach, tj. E_{kNN} (podejścia *wrapper* z klasyfikatorem kNN) oraz E_{DT} (podejścia *wrapper* z algorytmem indukcji drzew decyzyjnych). Każdy wiersz tabel opisuje przebieg eksperymentu przeprowadzonego z użyciem jednego z zestawów operatorów KI (Z_A, Z_B, Z_C), w tym:

- charakterystykę procesu KI: całkowitą liczbę ocenianych stanów $|F_{eval}|$ oraz liczbę wykonanych ruchów n_{mov},
- opis wyniku: liczność wyindukowanego zbioru cech $|F^+|$, rozkład długości cech $|f|$, końcową wartość funkcji oceniającej $E(F^+)$ w procentach,
- trafność klasyfikowania $\eta(T)$ uzyskiwaną z użyciem wygenerowanej reprezentacji (zbioru cech F^+) na zbiorze testującym T przez klasyfikatory wymienione w poprzednim podrozdziale.

Obliczenia wykonano na komputerze PC z procesorem Pentium II 400MHz z 128MB pamięci operacyjnej. Czas obliczeń realizowanych w ramach procesu KI

Z	Proces KI		Wynik KI						Trafność klas.		
	$\lvert F_{eval}\rvert$	n_{mov}	$\lvert F^+\rvert$	$\lvert f\rvert$				$E_{kNN}(F^+)$	$\eta(T)$ [%]		
				1	2	3	≥ 4	[%]	kNN	C4.5	SSN
A	21169	20	9	5	2	2		76.4	72.9	65.5	75.0
B	47592	20	15		15			79.4	75.0	73.2	77.3
C	29688	16	10	6	2	2		80.4	74.6	69.7	**77.8**

Tabela 6.1: Wyniki KI (funkcja oceniająca typu wrapper z klasyfikatorem kNN, E_{kNN}). Oznaczenia: Z - zestaw operatorów, $\lvert F_{eval}\rvert$ - liczba ocenianych stanów, n_{mov} - liczba wykonanych ruchów, $\lvert F^+\rvert$ - liczba cech w znalezionej reprezentacji suboptymalnej, $\lvert f\rvert$ - rozkład długości cech, $E_{kNN}(F^+)$ - wartość funkcji E_{kNN} dla znalezionego rozwiźzania

Z	Proces KI		Wynik KI						Trafność klas.		
	$\lvert F_{eval}\rvert$	n_{mov}	$\lvert F^+\rvert$	$\lvert f\rvert$				$E_{DT}(F^+)$	$\eta(T)$ [%]		
				1	2	3	≥ 4	[%]	kNN	C4.5	SSN
A	4865	8	6	4	2			59.6	61.0	60.8	66.8
B	34798	17	11		17			71.6	70.5	**74.4**	72.4
C	12269	10	6	3	2	1		62.8	59.7	61.5	66.2

Tabela 6.2: Wyniki KI (funkcja oceniająca typu wrapper z klasyfikatorem kNN, E_{DT})

wynosił od 50 do 140 minut. Wyniki prezentowane w tabelach 6.1 i 6.2 można podsumować w następujących punktach.

1. Algorytm SLS prowadzi do znalezienia zwięzłych reprezentacji F^+ (zbiory kilku do kilkunastu cech), które charakteryzują się jednak niezbyt wysoką trafnością klasyfikowania na zbiorze testującym, niezależnie od wykorzystywanego algorytmu indukcji.

2. Trafność klasyfikowania uzyskiwana z użyciem wyindukowanego zbioru cech jest pozytywnie skorelowana z jego rozmiarem $|F^+|$.

3. Końcowe wartości funkcji oceniającej $E(F^+)$ są bardziej trafne (w sensie: zgodne z trafnością klasyfikowania $\eta(T)$) w przypadku wykorzystywania podejścia wrapper z klasyfikatorem DT (E_{DT}), podczas gdy przy stosowaniu funkcji E_{kNN} są one nazbyt optymistyczne (zawyżone). Można z tego wnioskować, iż klasyfikator kNN ma w opisywanym zastosowaniu większe skłonności do przeuczenia.

Jednak ze względu na absolutne wartości $\eta(T)$, zbiory cech (reprezentacje) F^+ wygenerowane z użyciem funkcji oceniającej E_{DT} dają trafności klasyfikowania nieco gorsze od reprezentacji wyindukowanych z użyciem funkcji oceniającej E_{kNN}. Wynika to po części z faktu, że użycie E_{DT} prowadzi do mniej licznych zbiorów cech (por. wniosek 2 powyżej), co jest z kolei następstwem stromej heurystyki stosowanej w algorytmie indukcji drzewa decyzyjnego. Im lepsza indywidualna zdolność dyskryminacyjna cechy (mierzona za pomocą kryterium opartego na entropii, por. np. [Quinlan 1992]), tym większa szansa wykorzystania jej w węźle drzewa zbliżonym do korzenia. W konsekwencji drzewo wygenerowane na wczesnych etapach KI w oparciu o zestaw kilku dobrze dystryminujących cech rzadko kiedy podlega zasadniczej rewizji przy dodawaniu nowych cech. Dodawane cechy wykorzystywane są w budowaniu warunków w "wyższych" partiach drzewa, gdzie liczba przykładów jest już stosunkowo niewielka (co jest znaną wadą tej rodziny algorytmów indukcji), stąd trafność klasyfikowania tak rozbudowanego drzewa wzrasta w najlepszym przypadku nieznacznie, a często spada.

4. Znaczny wpływ na jakość uzyskiwanego rozwiązania (w sensie trafności klasyfikowania) ma rozmiar zbioru cech podstawowych F_0. Jeżeli jest on niewielki (jak w przypadku Z_A, gdzie zawiera 20 cech), algorytm SLS jest zmuszony do modyfikowania bieżącego rozwiązania poprzez operacje na pojedynczych operatorach KI, co prowadzi do rozwiązań zauważalnie gorszych od uzyskiwanych w przypadku, gdy zbiór F_0 jest bardziej liczny (np. Z_B, gdzie liczy 2600 cech).

Najważniejszy wniosek wypływający z prezentowanych wyników można ująć następująco: stosunkowo duży stopień trudności problemu (wynikający m.in. z obecności wielu klas decyzyjnych) w połączeniu z prostotą algorytmu przeglądania przestrzeni stanów (SLS), implikują tendencję procesu KI do "utykania" (po wykonaniu średnio zaledwie kilkunastu ruchów (n_{mov})) w stanach odpowiadających **lokalnym**

maksimum funkcji oceniającej E, i w konsekwencji, uzyskiwania niskiej trafności klasyfikowania $\eta(T)$. Jednocześnie pozytywna korelacja zarówno końcowej wartości funkcji oceniającej $E(F^+)$, jak i trafności klasyfikowania $\eta(T)$, z rozmiarem wyindukowanego zbioru cech $|F^+|$ sugeruje, iż przestrzeń stanów obfituje w wiele rozwiązań o lepszej wartości funkcji E, które są jednak nieosiągalne ze względu na stosowany algorytm przeszukiwania.

Konkluzja ta identyfikuje dwie główne przyczyny osiągania stosunkowo niskiej trafności klasyfikowania przez proponowane podejście: trudność (w tym wieloklasowość) problemu oraz "naiwny" charakter algorytmu przeszukiwania przestrzeni rozwiązań. W naturalny sposób prowadzi to do dwóch możliwości usprawnienia proponowanego podejścia, poprzez:

- wykorzystanie bardziej wyrafinowanych, w szczególności bardziej odpornych na problem optimów lokalnych, algorytmów przeszukiwania przestrzeni rozwiązań,
- dekompozycję oryginalnego wieloklasowego zadania uczenia maszynowego na podzadania charakteryzujące się mniejszymi ilościami klas decyzyjnych.

W ramach niniejszej pracy rozważono obie wyżej wymienione opcje, m.in. wykorzystując opisywany w punkcie 5.2.4 algorytm ewolucyjny. Wyniki przeprowadzonych eksperymentów zaprezentowane są w następnych podrozdziałach. Nie są to oczywiście jedyne możliwości modyfikacji proponowanej metody; inną drogą może być na przykład jednoczesne zastosowanie obu wyżej wymienionych środków, jednak podejście to wykraczałoby poza ramy niniejszej pracy.

Prezentowane wyżej wyniki sugerują także wyższość funkcji oceniającej E_{kNN} nad E_{DT} (w sensie trafności klasyfikowania $\eta(T)$ uzyskiwanej reprezentacji). Stąd eksperymenty opisywane w dalszych punktach pracy wykorzystują wyłącznie funkcję E_{kNN}.

6.3.2 Porównanie algorytmów przeszukiwania przestrzeni rozwiązań (SLS i AE)

Niniejszy punkt opisuje wybrane eksperymenty dotyczące porównania wyników osiąganych z użyciem poszczególnych algorytmów przeszukiwania przestrzeni rozwiązań: stromego przeszukiwania lokalnego SLS i algorytmu ewolucyjnego AE. Algorytmy ewolucyjne stosowano przy następujących ustawieniach parametrów:

- prawdopodobieństwo mutacji $p_{mut} = 0.05$,

Z	Wynik KI						Trafność klas.		
	$\lvert F^+\rvert$	$\lvert f\rvert$				$E(F^+)$	$\eta(T)$ [%]		
		1	2	3	≥ 4	[%]	kNN	C4.5	SSN
A	17	5	12			79.0	75.2	68.0	81.7
B	20		20			77.4	69.2	70.4	79.6
C	20	20	20			80.6	77.8	69.3	**83.2**

Tabela 6.3: Wyniki KI (algorytm AE). Oznaczenia: Z - zestaw operatorów, $|F^+|$ - liczba cech w znalezionej reprezentacji suboptymalnej, $|f|$ - rozkład długości cech, $E(F^+)$ - wartość funkcji E dla znalezionego rozwiźzania

Z	Wynik KI						Trafność klas.		
	$\lvert F^+\rvert$	$\lvert f\rvert$				$E(F^+)$	$\eta(T)$ [%]		
		1	2	3	≥ 4	[%]	kNN	C4.5	SSN
A	11	3	8			79.4	75.2	73.0	77.5
B	12	1	11			85.0	**80.5**	74.8	78.5
C	16	8	8			79.6	78.7	72.7	79.5

Tabela 6.4: Wyniki KI (algorytm AE z lokalną optymalizacją rozwiźzań)

- wielkość populacji $|P|_0 = 100$,
- początkowy rozmiar osobnika (liczba cech) $|F|_0 = 5$,

Pozostałe własności stosowanych algorytmów podane są w punkcie 5.2.4. Wykorzystano konwencjonalny algorytm ewolucyjny i algorytm ewolucyjny z lokalną optymalizacją rozwiązań (por. punkt 5.2.4). W drugim z wymienionych tu algorytmów do lokalnej optymalizacji poszczególnych osobników w P wykorzystano algorytm SLS, jednak ze względu na znaczną złożoność obliczeniową, lokalna optymalizacja nie była przeprowadzana dla wszystkich osobników z P, a jedynie kilku (5) najlepszych, tj. charakteryzujących się największą wartością funkcji przystosowania. Z tego samego względu ograniczono liczbę ruchów wykonywanych w ramach SLS do 1.

Tabele 6.3 i 6.4 prezentują wyniki eksperymentów odpowiednio dla konwencjonalnego algorytmu ewolucyjnego i algorytmu ewolucyjnego z lokalną optymalizacją rozwiązań (por. punkt 5.2.4). W tabelach, poza opisywanymi już wcześniej wielkościami, podano także numer pokolenia, w którym uzyskano osobnika reprezentującego najlepsze rozwiązanie F^+ (kolumna "Pokolenie"). Porównanie tych wyników z wynikami uzyskanymi z użyciem algorytmu stromego przeszukiwania lokalnego (Tabela 6.1) prowadzi do następujących konkluzji.

1. Wprowadzenie lokalnej optymalizacji rozwiązań do algorytmu ewolucyjnego prowadzi do zauważalnej poprawy wyników, tj. zauważalnego wzrostu końcowej wartości funkcji przystosowania E i trafności klasyfikowania $\eta(T)$, przy jednoczesnej znacznej (średnio niemal dwukrotnej) redukcji rozmiaru reprezentacji $|F^+|$.

2. Wyniki otrzymywane z użyciem algorytmów ewolucyjnych (AE) są znacząco lepsze od uzyskanych przy pomocy stromego przeszukiwania lokalnego (SLS). Dotyczy to zwłaszcza algorytmu ewolucyjnego z lokalną optymalizacją rozwiązań (Tabela 6.4), gdzie otrzymywane reprezentacje F^+ dają lepszą trafność klasyfikowania na zbiorze testującym T przy jednocześnie mniejszym lub porównywalnym rozmiarze reprezentacji $|F^+|$. Przyczyn tej poprawy należy dopatrywać się zwłaszcza w globalnym charakterze AE.

Poza tym należy zaznaczyć, iż, mimo że konwencjonalny algorytm ewolucyjny realizuje globalne przeszukiwanie przestrzeni rozwiązań, jego czasochłonność nie jest znacząco większa niż w przypadku wykorzystania SLS. Bierze się to stąd, iż każda iteracja konwencjonalnego algorytmu ewolucyjnego wymaga ocenienia funkcją oceniającą E wszystkich osobników z P, czyli $|P|$ rozwiązań (czyli np. zaledwie 100 dla wyżej wymienionych ustawień parametrów), co jest wielkością znacznie mniejszą od liczby stanów ocenianych w sąsiedztwie N w każdym kroku algorytmu SLS. Oczywiście w przypadku AE z lokalną optymalizacją rozwiązań złożoność ta jest większa.

6.3.3 Dekompozycja zadania KI na podzadania binarne

Jak wspomniano już w poprzednim podrozdziale, jednej z przyczyn stosunkowo niskiej trafności klasyfikowania uzyskiwanej przez proponowane podejście można doszukiwać się w obecności wielu klas decyzyjnych w analizowanym problemie. Zwłaszcza w przypadku stosowania algorytmu SLS trudno jest znaleźć zbiór cech dobrze dyskryminujący jednocześnie wszystkie klasy decyzyjne. Bieżące rozwiązanie F_k można poszerzyć o nową cechę f tylko wtedy, gdy rozbudowanie przestrzeni reprezentacji o ten dodatkowy *jednocześnie*

- polepszy rozpoznawanie przykładów klasyfikowanych dotychczas niepoprawnie, i
- nie pogorszy rozpoznawania przykładów klasyfikowanych dotychczas niepoprawnie.[2]

[2] Jest to przejrzysta, choć oczywiście nie do końca precyzyjna ilustracja słowna omawianego ograniczenia. W rzeczywistości, ponieważ trafność klasyfikowania ma charakter addytywny, między wymienianymi tu czynnikami zachodzi przetarg i dodanie cechy może zwiększać wartość funkcji

Gdy na przykład algorytmem indukcji stosowanym w metodzie *wrapper* jest algorytm kNN, przekłada się to na spełnienie pewnych ograniczeń dotyczących wzajemnych odległości pomiędzy przykładami. Wraz ze wzrostem wymiarowości przestrzeni reprezentacji (liczności zbioru cech) znalezienie cechy o podanych wyżej właściwościach staje się coraz trudniejsze. Problem ten zaznacza się szczególnie w obecności wielu (> 2) klas decyzyjnych, które dla zapewnienia odpowiedniej zdolności dyskryminacyjnej muszą tworzyć możliwie zwarte i parami rozłączne skupienia w przestrzeni reprezentacji.

Poszukując rozwiązania powyższego problemu należy dążyć do rozważania naraz możliwie małej ilości klas decyzyjnych. W szczególności, optymalnie jest rozważać najmiejszą możliwą liczbę klas, tj. dwie (C^+ i C^-), czyli tzw. *binarny* problem uczenia maszynowego (por. str. 32). Propozycje spotykanie najczęściej w literaturze (por. [Chan & Stolfo 1993], [Chan & Stolfo 1993]) sprowadzają się do jednego z dwóch następujących rozwiązań:

- dekompozycji n-klasowego problemu uczenia maszynowego na n problemów binarnych (por. np. [Dietterich & Bakiri 1995], ang. *one-of-n classifier*), przy czym klasy decyzyjne C^+ i C^- w k-tym problemie tworzone są według reguły:

$$C^+ = C_k,\ C^- = \bigcup_{k'=1,\, k' \neq k}^{n} C_{k'},$$

- dekompozycji n-klasowego problemu uczenia maszynowego na $\binom{n}{2} = n\,(n-1)\,/2$ problemów binarnych, tj. rozważanie wszystkich par klas (por. np. [Friedman 1996], [Chan & Stolfo 1993], [Jelonek, Krawiec, *et al.* 1998b], ang. n^2 *classifiers*, *pairwise classifiers*) przy czym klasy decyzyjne C^+ i C^- w problemie identyfikowanym parą indeksów $(k_1, k_2)\,, \quad k_1 < k_2, \quad k_1, k_2 \in \langle 1, n \rangle$tworzone są wg reguły:

$$C^+ = C_{k_1}, \quad C^- = C_{k_2}$$

Algorytmy uczenia maszynowego reprezentujące powyższą ideę nazywane są często w literaturze *metaklasyfikatorami* (ang. *metaclassifiers*), *klasyfikatorami złożonymi* lub *złożonymi systemami klasyfikującymi* [Jelonek & Stefanowski 1997], zaś poszczególne klasyfikatory wchodzące w ich skład - *klasyfikatorami bazowymi* (ang. *base classifiers*). W proponowanym podejściu wykorzystanie dekompozycji problemu oryginalnego sprowadza się do rozważenia wielu zadań konstruktywnej indukcji cech dla binarnych problemów uczenia maszynowego. Cechy wyindukowane dla poszczególnych podzadań mogą być następnie:

---

oceniającej E przy jednoczesnym pogorszeniu trafności klasyfikowania dla pewnego podzbioru przykładów.

Z	Wynik KI					Trafność klas. $\eta(T)$ [%]		
	$\lvert F^+\rvert$	$\lvert f\rvert$						
		1	2	3	≥ 4	kNN	C4.5	SSN
A	24	9	12	3		71.1	71.6	81.5
B	31	2	26	3		71.8	74.7	**82.7**
C	22	7	7	4	4	69.3	69.9	77.4

Tabela 6.5: Wyniki KI dla problemu zdekomponowanego na podproblemy binarne typu *1–z–n*. Oznaczenia: Z - zestaw operatorów, $|F^+|$ - liczba cech w znalezionej reprezentacji suboptymalnej, $|f|$ - rozkład długości cech

Z	Wynik KI					Trafność klas. $\eta(T)$ [%]		
	$\lvert F^+\rvert$	$\lvert f\rvert$						
		1	2	3	≥ 4	kNN	C4.5	SSN
A	119	45	59	11	4	81.1	74.2	85.6
B	218	21	184	12	1	82.4	80.3	**88.4**
C	112	41	48	11	12	82.6	77.2	86.7

Tabela 6.6: Wyniki KI dla problemu zdekomponowanego na podproblemy binarne typu n^2

- łączone w jeden zbiór cech (reprezentację), wykorzystywany dalej przez konwencjonalny algorytm indukcji, *lub*
- wykorzystywane w poszczególnych klasyfikatorach bazowych metaklasyfikatora.

Ponieważ metaklasyfikatory jako takie nie leżały w centrum zainteresowania niniejszej rozprawy, zastosowano pierwsze z wymienionych tu rozwiązań dla obu wcześniej wymienionych sposobów dekompozycji problemu wieloklasowego. Tabele 6.5 i 6.6 przedstawiają wyniki KI i testów z użyciem wybranych klasyfikatorów odpowiednio dla problemu zdekomponowanego w sposób "1 z n" i problemu zdekomponowanego w sposób n^2. Należy jednak zaznaczyć, iż w odróżnieniu od wcześniej prezentowanych tabel, wielkości podane w kolumnie "Wynik KI" dotyczą końcowego wyniku KI, tj. zbioru cech powstałego przez teoriomnogościowe zsumowanie reprezentacji otrzymanych w wyniku wszystkich binarnych procesów KI, których w pierwszym z omawianych przypadków jest 10, zaś w drugim $\binom{10}{2} = 45$.

Wyniki eksperymentów przeprowadzonych w ramach dekompozycji problemu na podproblemy binarne prowadzą do następujących wniosków.

1. Konstruktywna indukcja cech na problemie zdekomponowanym w sposób "1 z n" daje reprezentacje F^+ stosunkowo zwięzłe (mało liczne), ale jedynie nieznacz-

nie lepsze pod względem trafności klasyfikowania od rozwiązań uzyskiwanych bez dekompozycji problemu (por. np. tabela 6.1).

2. Dekompozycja problemu w sposób n^2 zapewnia zauważalną poprawę trafności klasyfikowania w porównaniu z rozwiązaniami uzyskiwanymi bez dekompozycji problemu przy użyciu algorytmu SLS. Poprawa ta jest jednak okupiona wielokrotnym wzrostem rozmiaru reprezentacji $|F^+|$ ($100 \div 200$ cech), który staje się wówczas porównywalny z liczbą punktów w analizowanym obrazie (24×24). Czytelność tak rozbudowanej reprezentacji pozostaje oczywiście pod znakiem zapytania.

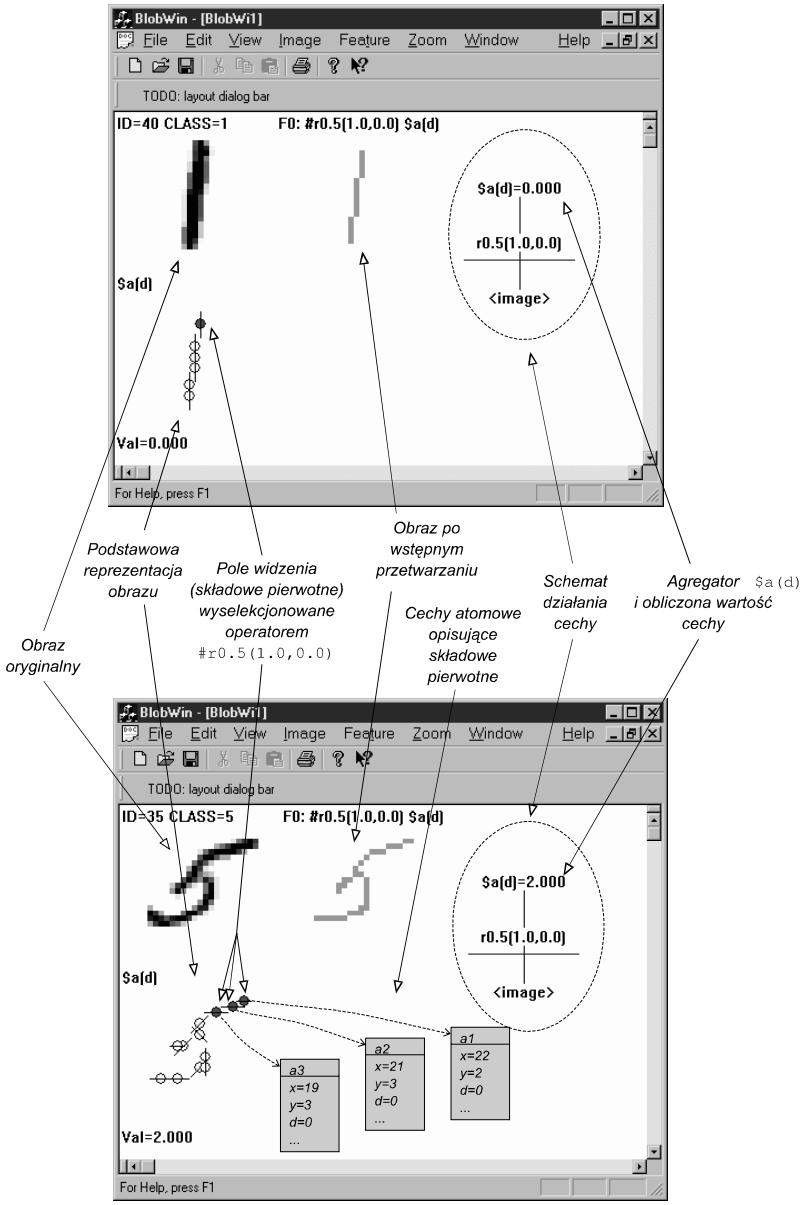
6.3.4 Interpretacja wybranego rozwiązania

Dla ilustracji działania metody przyjrzyjmy się zbiorowi cech wygenerowanemu w jednym z eksperymentów dla pary klas (C_1, C_5). Zawiera on trzy cechy, których reprezentacja w formacie zgodnym z implementacją komputerową (por. dodatek A) jest następująca:

```
Cecha 1: $a(d)
Cecha 2: #r0.5(1.0,0.0) $a(d)
Cecha 3: #r0.3(0.25,0.25) $n
```

Łatwo zaobserwować, iż pierwsza z obliczanych cech ma charakter globalny (nie wykorzystuje żadnych selektorów) i przy pomocy agregatora a obliczają średnią wartość cechy atomowej "kierunek" (d, *direction*). Druga z wyżej wymienionych cech oblicza tę samą wielkość ($\$a(d)$), ale w polu widzenia ograniczonym do kołowego pola recepcyjnego ($\#r$), którego środek mieści się w prawym górnym narożniku obrazu (współrzędne znormalizowane $(1.0, 0.0)$), a promień wynosi $0.\dot{5}$ (także we współrzędnych znormalizowanych). Ostatnia cecha operuje na kołowym polu widzenia o promieniu 0.3 i środku w punkcie $(0.25, 0.25)$, wyznaczając ilość ($\$n$) składowych pierwotnych w tym polu widzenia. Widać zatem, iż konstruktywna indukcja w proponowanym podejściu automatycznie wymusza dywersyfikację cech, co objawia się m.in. wyznaczaniem wartości poszczególnych cech na podstawie różnych fragmentów obrazu.

Rysunek 6.4 prezentuje proces obliczania drugiej z wymienionych cech dla przykładowych reprezentantów klas decyzyjnych rozważanych w tym zadaniu. Dla jednoczesnej ilustracji zaimplementowanego oprogramowania sposób obliczania cechy pokazany jest w oknie programu. Program, poza obrazem oryginalnym i wstępnie przetworzonym oraz jego reprezentacją podstawową, wyświetla także schemat graficzny cechy (graf) i wyróżnia ciemniejszymi kółkami składowe pierwotne (pole widzenia) wyselekcjonowane przez zaznaczony (podkreślenie) operator selekcji. Ostatni (najwyżej położony) węzeł w grafie reprezentującym cechę podaje także (poza opisem agregatora) obliczoną wartość cechy.



Rysunek 6.4: Ilustracja procesu obliczania wybranej cechy

6.4 Dyskusja wyników eksperymentów

Niniejszy podrozdział zawiera wnioski i obserwacje wynikające z wyników eksperymentów zaprezentowanych w poprzednim punkcie, pogrupowane według problemów których dotyczą.

Trafność klasyfikowania. W pracy [LeCun & et al. 1995] znaleźć można wyniki eksperymentów obliczeniowych przeprowadzanych na bazie danych MNIST wybranymi metodami uczenia maszynowego i rozpoznawania obrazów, w tym głównie nieliniowymi sieciami neuronowymi. Najlepsze wyniki ($95 \div 99\%$) uzyskiwane są przy użyciu specjalizowanych sieci neuronowych (np. LeNet), gdzie polepszenie wyniku uzyskano przez wzbogacenie zbioru uczącego o dodatkowe przykłady powstałe przez zniekształcenie przykładów oryginalnych. Stosunkowo dobry wynik uzyskuje także na "surowych", nieprzetworzonych obrazach metoda k najbliższych sąsiadów (kNN), mimo tego, że jest ona bardzo prosta i reprezentuje mało wyrafinowany paradygmat uczenia maszynowego, tzw. *lazy learning*. Szczegóły na temat stosowanych metod i wyników znaleźć można w pracach [LeCun & Bengio 1994], [LeCun & Bengio 1995], [LeCun & et al. 1995].

Porównanie wyników osiągniętych przez proponowane podejście z uzyskanymi innymi metodami prowadzi do konkluzji, że nie dorównuje im ono pod względem trafności klasyfikowania. Należy jednak podkreślić, iż **przedmiotem prowadzonych badań było przede wszystkim zbadanie możliwości automatycznego wyboru przestrzeni reprezentacji przez system uczący się** na podstawie informacji obrazowej, nie zaś opracowanie wyrafinowanej metody dedykowanej do konkretnego zastosowania, tj. rozpoznawania ręcznie pisanych znaków alfanumerycznych. W związku z tym zrezygnowano tu ze stosowania dodatkowych mechanizmów, które mogłyby zwiększyć uzyskiwaną trafność klasyfikowania, takich jak bardziej wyrafinowane przetwarzanie wstępne (np. normalizacja obiektów (cyfr) w sensie kąta nachylenia osi głównej (tzw. *deskewing*)), eliminacja artefaktów), czy też stosowanie zaawansowanych systemów klasyfikujących, np. metaklasyfikatorów (złożonych systemów klasyfikujących), metod hybrydowych, etc. Warto też zwrócić uwagę, że **wysoka trafność uzyskiwana przez podejścia bezpośrednie** (sieci neuronowe, podejścia minimalnoodległościowe) **w rozważanym zastosowaniu wynika po części z jego specyfiki**, na którą składa się m.in. niewielki rozmiar obrazu i jego "znormalizowany" charakter (m.in. w sensie braku problemu z niezmienniczością T, S i R, por. punkt 2.3).

W poniższych punktach wymieniono inne możliwe przyczyny uzyskiwania przez proponowane podejście rezultatów gorszych od uzyskiwanych metodami bezpośrednimi.

1. Większość algorytmów indukcji stosowanych w cytowanych wyżej pracach

prezentuje paradygmat podsymboliczny i nie pozyskuje wiedzy w postaci jawnej. Niektóre z nich w gruncie rzeczy nie przeprowadzają uczenia (w sensie indukcji), reprezentując metodologię *lazy learning* (np. metoda najbliższych sąsiadów). Inne natomiast (głównie sieci neuronowe) realizują adaptację do zbioru uczącego poprzez dostrajanie parametrów wielowymiarowej, złożonej funkcji nieliniowej. W przeciwieństwie do tych metod, **proponowane podejście prezentuje paradygmat symboliczny**, umożliwiając wgląd i rewizję pozyskanej wiedzy. Prowadzi to jednak jednocześnie do utraty pewnej elastyczności charakterystycznej np. dla sztucznych sieci neuronowych.

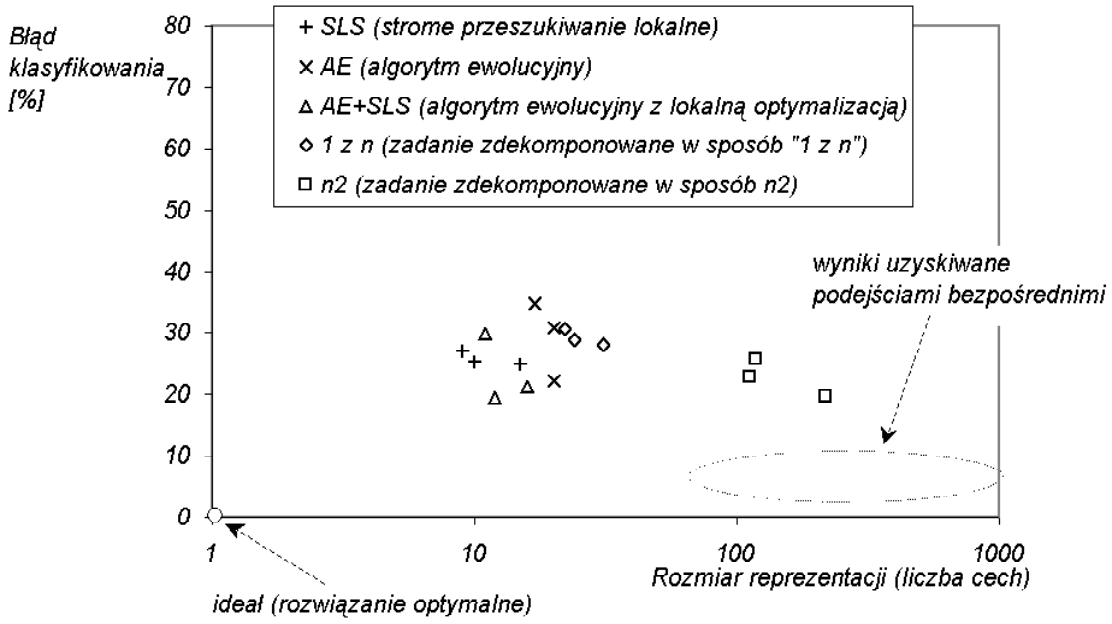
2. Ze względu na konieczność przeprowadzenia wielu eksperymentów, czas, jaki można było przeznaczyć na każdy z nich był dość ograniczony. Implikowało to użycie jako zbioru uczącego stosunkowo niewielkiego podzbioru obrazów zawartych w bazie MNIST. Należy się spodziewać, że użycie większych i w konsekwencji bardziej reprezentatywnych zbiorów uczących zapewniałoby lepsze rezultaty.

Rozmiar wygenerowanej reprezentacji. Przeprowadzone eksperymenty potwierdzają, że proponowana metoda jest zdolna do indukowania **bardzo zwięzłych reprezentacji obrazów zapewniających jednocześnie stosunkowo dużą zdolność rozpoznawania**. Reprezentacja składająca się z kilkunastu cech potrafi zapewnić trafność klasyfikowania w granicach 70-80%, co przy znacznej liczbie klas decyzyjnych (10) jest zdecydowanie interesującym wynikiem. Przyglądając się bliżej wygenerowanym reprezentacjom łatwo też zaobserwować, iż otrzymane cechy są stosunkowo proste; przeważają wśród nich cechy zawierające 1 do 3 opereatorów. Cechy o długości większej niż 4 należą do rzadkości.

Ta własność proponowanego podejścia owocuje następującymi zaletami:

- stosunkowo dobrą **czytelnością pozyskanej wiedzy** (zarówno w zakresie sposobu tworzenia reprezentacji, jak i klasyfikatora, o ile użyje się algorytmu indukcji reprezentującego paradygmat symboliczny),
- znaczną **szybkością procesu tworzenia reprezentacji** (obliczania wartości cech).

Rozwijając drugą z wymienionych tu zalet warto zwrócić uwagę, iż inne metody, o których mowa w wyżej cytowanych pracach, charakteryzują się znaczną złożonością procesu rozpoznawania. Dotyczy to zwłaszcza sztucznych sieci neuronowych, gdzie rozpoznanie wymaga propagacji pobudzenia (obrazu) przez sieć składającą się niekiedy z tysięcy sztucznych neuronów i setek tysięcy połączeń, jak i podejść minimalnoodległościowych, gdzie wysoki koszt obliczeniowy związany jest z koniecznością wyszukania w bazie danych (pamięci klasyfikatora) obrazu najbardziej podobnego do klasyfikowanego. Natomiast w proponowanym podejściu rozpoznawanie wymaga



Rysunek 6.5: Ilustracja przetargu pomiędzy rozmiarem reprezentacji a trafnością klasyfikowania na przykładzie uzyskanych wyników (trafność klasyfikowania dla klasyfikatora kNN)

obliczenia wartości zazwyczaj niewielkiej ilości prostych cech. Jeżeli do tego zastosuje się klasyfikator o niskiej złożoności obliczeniowej procesu klasyfikowania, można otrzymać system wnioskujący na podstawie informacji obrazowej charakteryzujący się znaczną szybkością rozpoznawania.

Między dwoma omawianymi wyżej kryteriami oceny systemów uczących się (trafnością klasyfikowania i rozmiarem reprezentacji) zachodzi oczywisty przetarg. Dla opisywanych eksperymentów przetarg ten zilustrowany jest na rysunku 6.5. Wynik ten przemawia za konkluzją, iż spośród wszystkich rozważanych w ramach tej rozprawy metaheurystyk, **najbardziej korzystne wyniki daje algorytm ewolucyjny wzbogacony o lokalną optymalizacją rozwiązań**.

Funkcja oceniająca. Algorytmy indukcji wykorzystane w funkcji oceniającej E nie są najbardziej wyrafinowanymi metodami uczenia maszynowego i nie są pozbawione pewnych wad. Metoda kNN nie ma możliwości różnicowania istotności poszczególnych atrybutów w różnych częściach przestrzeni cech, co w konsekwencji oznacza, że podczas klasyfikowania przykładu pod uwagę brane są wartości wszystkich cech. Z kolei hierarchiczna struktura drzew decyzyjnych powoduje, że budowa-

nie coraz głębszych węzłów w drzewie odbywa się na podstawie zmniejszającego się (i potencjalnie coraz mniej reprezentatywnego) zbioru przykładów. Rozbudowywanie drzewa jest jednocześnie niezbędne, bowiem dla n klas decyzyjnych liczba liści musi wynosić co najmniej n, co implikuje co najmniej $\lceil n \log_2 n \rceil$ wszystkich wezłów w drzewie.

Mimo zastosowania 4-krotnej walidacji skrośnej w funkcji oceniającej E, jej wartość końcowa $E\,(F^+)$ jest zazwyczaj znacząco wyższa niż uzyskiwana później trafność klasyfikacyjna na zbiorze testującym, stanowi zatem nazbyt optymistyczną estymatę zdolności predykcyjnej wyindukowanego zbioru cech F^+. Jest to objaw specyficznego przeuczenia charakterystycznego dla podejścia *wrapper*, w wyniku którego indukowany jest zbiór cech zapewniający możliwie dobrą trafność klasyfikowania η_{CV} w eksperymencie walidacji skrośnej przy pewnym *ustalonym podziale na podzbiory* T_l (por. punkt 4.2.4; stosowanie stałego podziału na podzbiory T_l jest niezbędne dla zapewnienia porównywalności wartości funkcji E dla różnych podzbiorów cech). Zjawisku temu można by zapobiec stosując większą liczbę powtórzeń n_{CV} lub powtarzając wielokrotnie całą walidację skrośną, jednak zwiększyłoby to wielokrotnie czas trwania obliczeń. Poza tym wydaje się, że bezwzględna wartość funkcji oceniającej E nie jest aż tak ważna; bardziej istotne jest, aby prawidłowo różnicowała ona poszczególne zbiory cech.

Ponadto warto też zwrócić uwagę, iż **sztuczne sieci neuronowe potwierdziły swoją przydatność w zadaniu klasyfikacji**, uzyskując (z nielicznymi wyjątkami) najlepszą trafność klasyfikowania spośród wykorzystywanych klasyfikatorów. Jest to obserwacja warta podkreślenia, ponieważ należy pamiętać, iż algorytmem indukcji stosowanym w podejściu *wrapper* (funkcja oceniająca E) był klasyfikator minimalnoodległościowy kNN, tak więc cały proces poszukiwania suboptymalnej reprezentacji uwzględniał ukierunkowanie indukcyjne właśnie tego algorytmu. Mimo to sztuczna sieć neuronowa wykazała się lepszą zdolnością tworzenia przydatnych uogólnień.

Rozdział 7

Przykład zastosowania podejścia w diagnostyce nowotworów OUN

Niniejszy rozdział opisuje zastosowanie proponowanego podejścia do wspomagania diagnozowania nowotworów ośrodkowego układu nerwowego (OUN) na podstawie obrazów mikroskopowych skrawków histologicznych. Rozdział otwiera krótkie wprowadzenie w zagadnienie patomorfologii OUN, a następnie opisuje w jaki sposób proponowana metoda została dostosowana do tego problemu. Dalej następuje opis analizowanego zbioru przykładów i sposobu przeprowadzenia eksperymentów obliczeniowych oraz prezentacja wyników i analiza wybranej reprezentacji otrzymanej w wyniku konstruktywnej indukcji cech obrazu. Wątkiem szczególnie silnie akcentowanym w niniejszym rozdziale jest demonstracja możliwości wyjaśniania decyzji podejmowanych przez system wspomagania decyzji w ramach proponowanego podejścia.

7.1 Patomorfologia OUN

Informacja obrazowa odgrywa bardzo istotną rolę w diagnostyce medycznej. W praktyce wiele decyzji, zwłaszcza dotyczących rozpoznania, podejmowanych jest między innymi na podstawie obrazów otrzymanych w wyniku badań makroskopowych (konwencjonalnych prześwietleń rentgenowskich, badań ultrasonograficznych, tomografii rentgenowskiej transmisyjnej, tomografii magnetycznego rezonansu ją drowego, itp.) i mikroskopowych (mikroskopia optyczna konwencjonalna, kontrastowo-fazowa, itp.). W literaturze przedmiotu spotkać można opisy wielu zastosowań metod przetwarzania i analizy obrazu do wspomagania procesu diagnozowania (por. np. [Shortliffe, Perreault, *et al.* 1990], [Kulikowski 1993], [Marchevsky & Bartels 1994], [Kącki 1997], [Jelonek, Krawiec, *et al.* 1998a], [Szymaś 1997]).

Niniejszy rozdział opisuje zastosowanie podejścia opisanego rozdziale 5 do wspomagania diagnozowania nowotworów ośrodkowego układu nerwowego (OUN) na podstawie obrazów mikroskopowych skrawków histologicznych. Na wybór tego zastosowania złożyły się m.in. następujące przesłanki:

- Jest to w praktyce patologicznej zagadnienie szczególnie trudne, wymagające długiego kształcenia patologa, co wynika m.in. ze znacznej różnorodności form nowotworów OUN (w praktyce występuje kilkadziesiąt ważnych typów nowotworów OUN).
- Uwzględnienie informacji obrazowej (tu: preparatu histologicznego) w procesie diagnozowania jest w wielu przypadkach nieodzowne, gdyż inne dane o pacjencie (np. wywiad kliniczny, wyniki badań tomograficznych) nie są wystarczające do dyskryminowania pomiędzy pewnymi typami nowotworów.
- W środowisku medycznym brak jest powszechnie przyjętych standardów w diagnozowaniu nowotworów OUN, co pociąga za sobą znaczną indywidualizację podejść wypracowywanych przez poszczególnych specjalistów i, w konsekwencji, rozbieżność rozpoznań przy diagnozowaniu trudniejszych przypadków.
- Poprawność rozpoznania typu nowotworu i jego stopnia złośliwości może mieć decydujące znaczenie dla dalszej terapii (np. farmakologicznej, chirurgicznej, chemicznej, radiacyjnej) stosowanej wobec pacjenta.

7.2 Wykorzystanie proponowanego podejścia

Celem prowadzonych wcześniej przez nas prac badawczych związanych z tym zagadnieniem [Jelonek & Krawiec 1993], [Jelonek, Krawiec, *et al.* 1994a], [Jelonek, Krawiec, *et al.* 1995], [Jelonek, Krawiec, *et al.* 1997], [Jelonek, Krawiec, *et al.* 1998a], [Jelonek, Krawiec, *et al.* 1998c], [Jelonek, Krawiec, *et al.* 1999], [Komosiński & Krawiec 2000] było zbudowanie systemu wspomagania diagnozowania o możliwie wysokiej skuteczności diagnostycznej. W tym celu konieczne było zaimplementowanie w nim wiedzy dotyczącej dziedziny zastosowania, co napotyka na, omawiane w rozdziale 4, trudności dotyczące procesu jej pozyskiwania od eksperta na drodze konwencjonalnego dialogu. W rozważanym zastosowaniu dodatkowym utrudnieniem staje się też wyżej wymieniony brak powszechnie przyjętych standardów w procesie diagnozowania, co utrudnia wykorzystanie innych niż ekspert źródeł wiedzy (np. podręczników).

Stąd w praktyce jedyną drogą pozyskania wiedzy o problemie jest **uczenie się z przykładów poprawnie zdiagnozowanych przez eksperta**, bazując na różnych

Astrocytoma anaplasticum	*Astrocytoma fibrillare*
Astrocytoma fibrillare	*Oligodendroglioma isomorphum*
Astrocytoma gemistocyticum	*Ependymoma*
Astrocytoma pilocyticum	*Choroid plexus papilloma*
Astrocytoma protoplasmaticum	*Glioblastoma multiforme*
Glioblastoma multiforme	*Medulloblastoma*

Tabela 7.1: Klasy nowotworów reprezentowane w analizowanym zbiorze przykładów

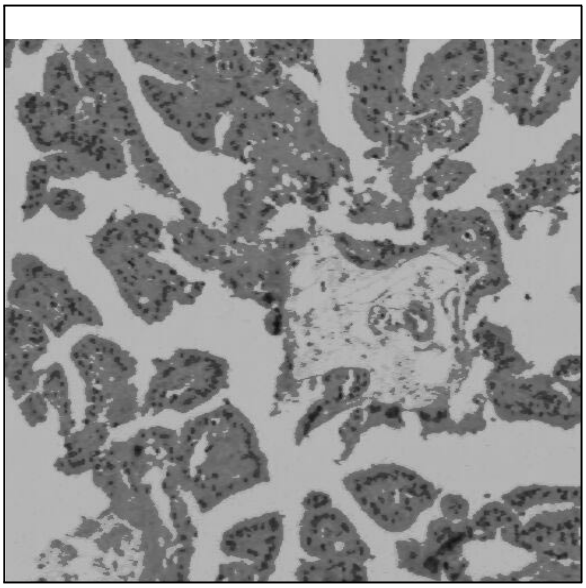
typach cech obrazowych (np. histogramach, cechach teksturalnych, cechach strukturalnych, itp.) i systemach uczących się działających w ramach różnych paradygmatów (podejścia minimalnoodległościowe, reguły i drzewa decyzyjne, sztuczne sieci neuronowe, por. rozdział 4). Wyniki otrzymywane przy użyciu podejść konwencjonalnych, prezentowane w wyżej cytowanych pracach, nie były jednak satysfakcjonujące z medycznego punktu widzenia: trafność klasyfikowania nie przekraczała 50..60%, w zależności od doboru cech i klasyfikatorów. Wyniki te sugerowały, iż rozważane zastosowanie należy do trudnych i znacząco odbiegały od określanych w literaturze jako minimum 80% (por. np. [Bouckaert 1988]), które system musi zapewniać, aby można było mówić o jakimkolwiek wspomaganiu diagnozowania.

Inną wadą stosowanych podejść konwencjonalnych była trudność interpretacji cech obrazu wykorzystywanych we wnioskowaniu, co utrudniało, kluczowe w zastosowaniach medycznych, wyjaśnianie decyzji podejmowanych przez system. Wreszcie proces projektowania i implementacji komputerowej cech, zwłaszcza tych dedykowanych do zastosowania, był żmudny i bazował jedynie na subiektywnych domysłach projektanta podpartych sugestiami eksperta. Obserwacje te stanowiły główne przesłanki dla wykorzystania proponowanego podejścia w tym zastosowaniu.

7.2.1 Zbiór przykładów

Analizowany zbiór obrazów reprezentuje wybraną grupę nowotworów OUN, tzw. nowotwory neuroepitelialne (nabłonkowe). W skład tej grupy wchodzi 14 typów guzów, głównie gwiadziaków (łac. *astrocytoma*), o różnych stopniach złośliwości (I...IV, [Kleihues, Buerger, *et al.* 1993]). Tabela 7.1 prezentuje listę klas nowotworów reprezentowanych w rozważanym zbiorze.

Klasy decyzyjne są równoliczne, każda z nich reprezentowana jest przez 5 przypadków (pacjentów). Dla każdego pacjenta pozyskano 10 obrazów ze skrawka (preparatu) histologicznego, tj. bardzo cienkiej (kilkanaście mikronów) warstwy materiału pozyskanego w drodze badania śródoperacyjnego lub biopsji cienkoigłowej. Bazę danych stanowi zatem 700 ($= 14\,\text{klas} \times 5\,\text{przypadków} \times 10\,\text{obrazów}$) obra-



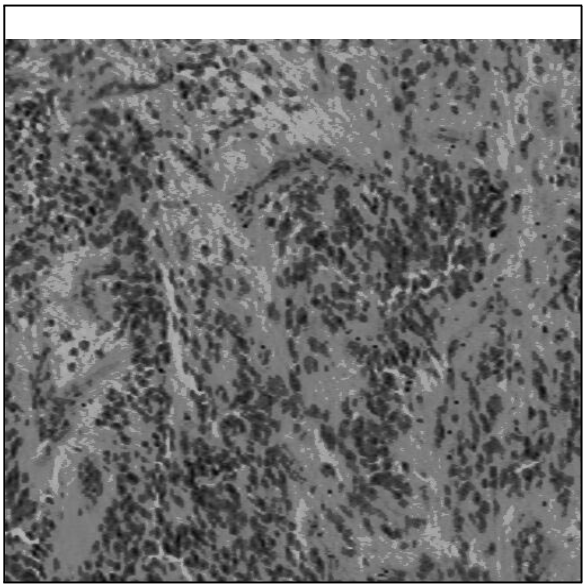
Rysunek 7.1: Przykładowy obraz preparatu histologicznego nowotworu OUN

zów mikroskopowych skrawków histologicznych. Każdy obraz reprezentuje inny fragment preparatu pozyskany przy użyciu mikroskopu optycznego przy średnim powiększeniu (ok. 100×). Obrazy pozyskiwane były przy użyciu kamery CCD zainstalowanej na mikroskopie. Każdy obraz ma wielkość 512×512 punktów, przy 24-bitowej skali kolorów i przechowywany jest w bezstratnym formacie rastrowym (*Windows Bitmap*, BMP) dla zapewnienia maksymalnej wierności w stosunku do obrazu oryginalnego. Przykładowe obrazy reprezentujące poszczególne typy nowotworów pokazują Rys. 7.1 i 7.2.

7.2.2 Podstawowa reprezentacja obrazu i operatory KI

Jak można zaobserwować na Rys. 7.1 i 7.2, w analizowanych obrazach skrawków histologicznych OUN przy zastosowanym powiększeniu mikroskopu widoczne są jądra komórkowe, drobne naczynia krwionośne, zwapnienia, itp. Poszczególne elementy (zwłaszcza jądra komórkowe) są na tyle drobne, że ich dokładna analiza morfologiczna, którą stosuje się często w pokrewnych zastosowaniach, jest utrudniona bądź nawet niemożliwa.

Niska rozdzielczość przestrzenna jest jednak wynikiem nieuniknionego kompromisu z wielkością fragmentu preparatu pokrywanego przez pole widzenia mikroskopu. Przy ustalonej rozdzielczości kamery, uzyskanie większej ilości detali wy-



Rysunek 7.2: Przykładowy obraz preparatu histologicznego nowotworu OUN

magałoby zawężenia pola widzenia do znacznie mniejszego fragmentu preparatu. Byłoby to jednak niekorzystne, ponieważ z sugestii eksperta medycznego wynika, że w diagnozowaniu nowotworów OUN na podstawie preparatów histologicznych szczególnie istotne znaczenie ma tzw. *architektonika* preparatu, tj. przestrzenny układ jąder komórkowych. Słuszność tej sugestii została ponadto potwierdzona w niezależnych badaniach, gdzie przeprowadzono analizę istotności poszczególnych atrybutów *opisu obrazu* stworzonego przez eksperta [Jelonek & Krawiec 1993] (z wykorzystaniem teorii zbiorów przybliżonych [Pawlak 1982]). Także inne eksperymenty prowadzone na tym samym zbiorze obrazów, wykorzystujące przebieg procesu segmentacji obrazu jako źródło informacji o jego strukturze [Komosiński & Krawiec 2000], potwierdziły tę obserwację.

W związku z taką charakterystyką analizowanych obrazów, dogodną reprezentacją podstawową wydaje się być *graf przyległości obszarów/regionów* (GPO, ang. *region adjacency graph*, por. np. [Pavlidis 1982], rozdział 6). W grafie takim wierzchołki reprezentują regiony (spójne/połączone zbiory punktów obrazu), zaś krawędzie - relację sąsiedztwa pomiędzy nimi. Do otrzymania podziału obrazu na regiony konieczne jest przeprowadzenie segmentacji obrazu (por. rozdział 2), której celem jest wyróżnienie w obrazie regionów o jednolitej charakterystyce (kolorystycznej, faktury, itp.). W literaturze opisuje się wiele metod segmentacji (por. np. [Haralick & Shapiro 1985]), które można uporządkować w trzy grupy: segmentację przez progo-

wanie (ang. *thresholding*), segmentację przez wykrywanie krawędzi (ang. *boundary detection*) oraz segmentację przez rozrost obszaru (ang. *region growing*). W opisywanym zastosowaniu użyto algorytmu segmentacji obrazu przez rozrost obszaru (por. np. [Zucker 1976]), jako najbardziej odpowiedniego ze względu na charakterystykę analizowanych obrazów. Algorytmy tego typu tworzą regiony w obrazie poprzez stopniowe łączenie mniejszych obszarów w większe. W szczególności, ponieważ rozważane zastosowanie nie narzuca wygórowanych ograniczeń na czas trwania procesu rozpoznawania, zastosowano algorytm charakteryzujący się stosunkowo wysoką złożonością obliczeniową, ale dający w wyniku obraz (zbiór regionów) bardzo wierny w stosunku do obrazu oryginalnego.

W rozważanym zastosowaniu reprezentacja podstawowa obrazu $r(x)$ tworzona była poprzez utożsamienie wierzchołków GPO ze składowymi pierwotnymi a_i. Każda składowa pierwotna opisana jest następującymi cechami atomowymi A_0 charakteryzującymi skojarzony z nią region (w nawiasach prostokątnych podano symbole stosowane w implementacji komputerowej, patrz dodatek A)[1]:

- polem powierzchni [A],
- wartościami składowych podstawowych koloru w przestrzeni RGB [cR, cG, cB].

Zbiór S operatorów selekcji pola widzenia określony jest w sposób bardzo zbliżony do zbioru wykorzystywanego w zastosowaniu do rozpoznawania znaków (por. punkt 6.2.2): selektory dokonują selekcji pola widzenia poprzez narzucanie ograniczeń na wartości cech atomowych. Zbiór stosowanych agregatorów A był dokładnie taki sam, jak omawiany w podrozdziale 6.2.2.

7.2.3 Wyniki eksperymentów obliczeniowych

Tabela 7.2 prezentuje wyniki eksperymentów obliczeniowych przeprowadzonych dla wybranych ustawień parametrów, tj.

- algorytmów przeszukiwania przestrzeni reprezentacji (stromego przeszukiwania lokalnego SLS, zachłannego przeszukiwania lokalnego[2] (ang. *greedy local search, GLS*), i algorytmów ewolucyjnych AE),

[1] Rozważano też inne cechy atomowe, w tym opis regionu w przestrzeni reprezentacji barw HSI oraz liczbę regionów sąsiednich (przyległych w GPO), które nie wpływały jednak znacząco na jakość otrzymywanych wyników.

[2] GLS różni się od SLS wybieraniem z sąsiedztwa pierwszego napotkanego stanu zwiększającego wartość funkcji oceniającej E.

Algorytm KI	Wynik KI						Trafność klas.		
	$\|F^+\|$	$\|f\|$				$E(F^+)$	$\eta(T)$ [%]		
		1	2	3	≥ 4	[%]	kNN	C4.5	SSN
SLS+E_{kNN}	7	2	2	2	1	75.4	75.4	60.0	62.8
GLS+E_{kNN}	8					73.5	**78.2**	62.1	61.8
SLS+E_{DT}	3	1			2	39.0	54.3	42.1	55.3
AE+E_{kNN}	6	3	2	1		71.2	75.7	57.9	62.5

Tabela 7.2: Wyniki KI w zastsosowaniu do rozpoznawania nowotwor ów OUN. Oznaczenia: $|F^+|$ - liczba cech w znalezionej reprezentacji suboptymalnej, $|f|$ - rozkład długości cech, $E(F^+)$ - wartość funkcji E dla znalezionego rozwiązania (E_{kNN} - metoda wrapper z klasyfikatorem kNN, E_{kNN} - metoda wrapper z drzewem decyzyjnym)

- funkcji oceniającej (metoda *wrapper* z klasyfikatorem minimalnoodległościowym E_{kNN} i podejście *wrapper* z klasyfikatorem typu drzewo decyzyjne E_{DT}).

Eksperymenty przeprowadzane były, podobnie jak w zastosowaniu do rozpoznawania ręcznie pisanych znaków, przy jednokrotnym podziale na zbiór uczący i testujący. Zbiór uczący L składał się z wybranych losowo 60% obrazów (420 przykładów), zaś testujący T z pozostałych 280 obiektów. Do konstruktywnej indukcji cech obrazu wykorzystywane były oczywiście **jedynie przykłady ze zbioru uczącego** L. Po zakończeniu procesu konstruktywnej indukcji cech obrazu przeprowadzano eksperyment uczenia na zbiorze L i testowania na zbiorze T wybranych algorytmów uczenia maszynowego, reprezentujących różne metodologie (ostatnie trzy kolumny tabeli 7.2).

Zważywszy na znaczną liczbę klas decyzyjnych obecnych w zbiorze przykładów (14), **uzyskiwane trafności klasyfikowania należy uznać za wysokie**, trafność klasyfikowania klasyfikatora "losowego" przy zrównoważonych licznościach klas decyzyjnych wynosi bowiem zaledwie $\frac{100\%}{14} \cong 7.1\%$. Wyniki otrzymywane przy pomocy proponowanego podejścia są także porównywalne z otrzymywanymi na tym samym zbiorze obrazów z wykorzystaniem podejść konwencjonalnych (por. [Jelonek, Krawiec, *et al.* 1997], [Jelonek, Krawiec, *et al.* 1998a], [Jelonek, Krawiec, *et al.* 1998c], [Jelonek, Krawiec, *et al.* 1999]), charakteryzują się jednak **zdecydowanie mniejszą liczbą cech ekstrahowanych z obrazu**. Ponadto otrzymane cechy są łatwiej interpretowalne dla eksperta dziedziny zastosowania (patologa), niż otrzymywane w podejściach konwencjonalnych, które często mają bardziej "techniczny" charakter (np. histogramy). W następnym punkcie wybrany wynikowy zbiór cech zaprezentowany zostanie w szczegółach. W szczególności, interesujący jest wynik eksperymentu wykorzystującego drzewo decyzyjne w funkcji oceniającej

Cecha	Definicja	Długość
f_0	#<(cG,0.25) $a(cB)	2
f_1	#>(cG,0.1) $d(cB)	2
f_2	$a(cR)	1
f_3	#<(cB,0.5) #>(cB,0.33) $a(cG)	3
f_4	#<(cG,0.25) #<(cR,0.66) $d(cR)	3
f_5	#>(cR,0.25) $d(cB)	2
f_6	$a(cG)	1
f_7	#<(cG,0.5) #<(cR,0.66) $a(cR)	3

Tabela 7.3: Reprezentacja wygenerowana przy pomocy algorytmu zachłannego przeszukiwania lokalnego

(SLS+E_{DT}), gdzie stosunkowo niska trafność klasyfikowania zrekompensowana jest zwięzłością opisu (zaledwie trzy cechy).

Wyniki prezentowane w tabeli 7.2 ilustrują także znaczny wpływ ukierunkowania indukcyjnego wnoszonego przez funkcję oceniającą na trafność klasyfikowania uzyskiwaną na zbiorze testującym z wykorzystaniem wyindukowanej reprezentacji. Stąd bierze się w prezentowanych wynikach przewaga na korzyść, bardzo prostego przecież, algorytmu minimalnoodległościowego kNN i, dość zaskakujące, niskie trafności klasyfikowania uzyskiwane przez sztucznie sieci neuronowe (SSN)[3].

7.2.4 Analiza wybranej reprezentacji

Medycyna należy do tych pól zastosowań sztucznej inteligencji, w których aspekt wyjaśniania (uzasadniania) podejmowanych decyzji ma szczególne znaczenie. Niniejszy podrozdział ilustruje tę zdolność proponowanego podejścia na przykładzie repezentacji otrzymanej w wyniku zastosowania zachłannego algorytmu przeszukiwania lokalnego (wiersz oznaczony GLS+E_{kNN} w tabeli 7.2). Wyindukowana przestrzeń reprezentacji składa się z 8 cech, których definicje zaprezentowane są w tabeli 7.3 (zgodnie z notacją opisaną w dodatku A).

Omawiana przestrzeń reprezentacji jest wynikiem konstruktywnej indukcji cech obrazu przeprowadzonej z wykorzystaniem metody *wrapper* w charakterze funkcji oceniającej (E_{kNN}). Algorytm kNN wykorzystywany tu jako algorytm induk-

[3] Głębszą przyczyną jest tu zapewne pewna "niekompatybilność" sposobów podziału przestrzeni przykładów w algorytmie kNN i w SSN z neuronami obliczającymi pobudzenie jako sumę ważoną wejść. Należy przypuszczać, że wykorzystanie SSN o innej charakterystyce neuronów (ang. *post synaptic potential function, PSP*), np. sieci z neuronami o symetrii kołowej (RBF), przyniosłoby lepsze rezultaty.

cji wnosi pewne **ukierunkowanie indukcyjne** w przeszukiwaniu przestrzeni stanów (por. punkt 5.3), co implikuje, iż najbardziej naturalne byłoby wykorzystanie tego właśnie induktora do eksperymentów opisywanych w dalszej części tego podrozdziału. Ponieważ jednak kNN nie oferuje praktycznie żadnych możliwości opisu pozyskanej wiedzy czy wyjaśniania podejmowanych decyzji (poza powołaniem się na przykład/przykłady najbliższe ze względu na stosowaną miarę odległości), trzeba zdać się na inny typ algorytmu indukcji, charakteryzujący się jawną reprezentacją wiedzy[4].

Dlatego jako formę (język) reprezentacji wiedzy przyjęto w poniższych eksperymentach **reguły decyzyjne**. Za tym wyborem przemawia jej czytelność oraz zdolność do opisywania wyjątków, które są na przykład kłopotliwe dla alternatywnego podejścia symbolicznego, tj. drzew decyzyjnych [Breiman, Friedman, *et al.* 1984], [Quinlan 1992]. Ponadto pewną rolę odegrał tu stosunkowo duży stopień akceptacji tej reprezentacji wiedzy w środowisku medycznym.

Do indukowania reguł wykorzystano algorytm indukcji *Explore* oparty na teorii zbiorów przybliżonych (ang. *rough set theory*, [Pawlak 1982], [Pawlak 1991]), zaimplementowany w środowisku ROSE [Prędki, Słowiński, *et al.* 1998]. Reprezentuje on grupę algorytmów indukcji wymagających, aby wszystkie cechy (atrybuty warunkowe) miały dyskretny zbiór wartości (czyli były określone na skali porządkowej lub nominalnej, por. punkt 4.2.2). Przed zastosowaniem algorytmu indukcji do zbioru przykładów niezbędne jest zatem przeprowadzenie *dyskretyzacji* wartości atrybutów. Przeprowadzono ją przy pomocy popularnej metody Fayyada-Iraniego [Fayyad & Irani 1992], także zaimplementowanej w środowisku ROSE. Jest to algorytm dyskretyzacji *lokalnej*, tj. przetwarzający poszczególne atrybuty niezależnie.

Dyskretyzacja doprowadziła do wyróżnienia na skalach poszczególnych atrybutów przedziałów zaprezentowanych w tabeli 7.4. Warto podkreślić, iż ponieważ wykorzystywany algorytm dyskretyzacji heurystycznie poszukuje podziału skali oryginalnego atrybutu ciągłego na podprzedziały z wykorzystaniem kryterium zawartości informacyjnej (entropii), już **wynik dyskretyzacji w połączeniu z definicjami cech może stanowić istotne źródło wiedzy o rozważanym zastosowaniu**. Przedziały dyskretyzacji wyróżniają bowiem pewne **typy składowych pierwotnych** (tu: regionów w GPO) ze względu na wartości wyindukowanych cech. I tak na przykład wynik dyskretyzacji atrybutu f_0 sugeruje wyróżnienie dwóch typów regionów ze względu na wartość tej cechy.

Z racji tego, iż celem prezentowanych tu eksperymentów było pozyskiwanie wie-

[4]Z podobnej przyczyny nie rozważamy tu sztucznych sieci neuronowych (SSN), choć ostatnie dwie dekady obfitowały w prace prezentujące metody ekstrakcji wiedzy ze SSN lub zmierzające do wyjaśniania decyzji podejmowanych przez sieć (por. np. [Diederich 1992], [Fu 1994], [Craven & Shavlik 1994], [Krawiec, Słowiński, *et al.* 1998], [Faifer, Janikow, *et al.* 1999]).

Cecha	Przedziały dyskretyzacji
f_0	$(-\infty,0.48)$ $<0.48,+\infty)$
f_1	$(-\infty,0.05)$ $<0.05,0.07)$ $<0.07,0.08)$ $<0.08,+\infty)$
f_2	$(-\infty,0.32)$ $<0.32,0.37)$ $<0.37,0.46)$ $<0.46,0.55)$ $<0.55,+\infty)$
f_3	$(-\infty,0.15)$ $<0.15,0.26)$ $<0.26,0.30)$ $<0.30,0.34)$ $<0.34,+\infty)$
f_4	$(-\infty,0.05)$ $<0.05,0.08)$ $<0.08,0.10)$ $<0.10,+\infty)$
f_5	$(-\infty,0.02)$ $<0.02,0.03)$ $<0.03,0.06)$ $<0.06,+\infty)$
f_6	$(-\infty,0.12)$ $<0.12,0.15)$ $<0.15,0.21)$ $<0.21,0.27)$ $<0.27,0.28)$ $<0.28,+\infty)$
f_7	$(-\infty,0.32)$ $<0.32,0.46)$ $<0.46,0.55)$ $<0.55,+\infty)$

Tabela 7.4: Przedziały otrzymane w wyniku dyskretyzacji atrybutów z tabeli 7.3

Nr	Reguła	Pokrycie	Wsparcie
1	$(f_2 = 0)$ & $(f_6 = 3)$ => (D=9)	15	15
2	$(f_6 = 3)$ & $(f_7 = 0)$ => (D=9)	15	15
3	$(f_2 = 4)$ => (D=10)	24	23
4	$(f_7 = 3)$ => (D=10)	25	24

Tabela 7.5: Zbiór reguł wyindukowanych dla reprezentacji z tabeli 7.3, przy poziomie dyskryminacji 95% i minimalnej sile reguły 10 (D – atrybut decyzyjny, *Pokrycie* – liczba wszystkich przykładów ze zbioru uczącego, dla których spełniona jest przesłanka reguły, *Wsparcie* – liczba przykładów pokrytych należą cych do klasy decyzyjnej wskazywanej przez decyzję reguły)

dzy, a nie bezwzględnie dążenie do maksymalizacji trafności klasyfikowania, zdecydowano się na generowanie tzw. *opisu satysfakcjonującego*, tj. nie wymagającego stuprocentowej zdolności dyskryminacyjnej poszczególnych reguł oraz nie zakładającego, że wszystkie przykłady ze zbioru uczącego muszą być pokryte przez co najmniej jedną regułę. Ujmując to precyzyjniej, każda generowana reguła musi spełniać ograniczenia narzucone na następujące trzy parametry:

- **poziom dyskryminacji** (ang. *discrimination level*) - minimalny procent przykładów pokrywanych przez regułę, należących do wskazywanej przez nią klasy decyzyjnej,
- (bezwzględna) **siłę reguły** (ang. *minimal strength*) - minimalna liczba przykładów pokrywanych przez regułę,
- **długość reguły** - maksymalna liczba warunków elementarnych, czyli selektorów.

We opisywanych niżej eksperymentach stosowano maksymalną długość reguły równą 10. Zbiór uczący L stanowiły przykłady wykorzystywane wcześniej w procesie konstruktywnej indukcji (60% zbioru, 420 obrazów), zaś na zbiór testujący T składało się pozostałych 280 obrazów, opisanych obliczonymi dla nich ośmioma cechami prezentowanymi w tabeli 7.3).

Tabela 7.5 prezentuje zbiór reguł wygenerowany algorymem *Explore* **przy poziomie dyskryminacji 95%, i minimalnej sile reguły równej 15**. Składa się on zaledwie z 4 reguł, wskazujących na klasy decyzyjne 9 i 10. Oznacza to, iż tylko dla tych klas decyzyjnych dało się wygenerować reguły spełniające narzucone ograniczenia. Taki zbiór reguł nie może oczywiście służyć do klasyfikowania, ale stanowi on dobrą ilustrację najsilniejszych wzorców zawartych w informacji obrazowej.

Na szczególną uwagę zasługują reguły nr 3 i 4, z racji

- bardzo dużego pokrycia (odpowiednio 24 i 25 przykładów),
- bardzo wysokiej zdolności dyskryminacyjnej (odpowiednio 95.8% i 96.0%),
- prostoty (obie reguły zawierają zaledwie po jednym warunku elementarnym w części warunkowej).

Reguła nr 3 jest stosunkowo prosta, ponieważ opiera się na zaledwie jednej przesłance (selektorze), wykorzystującej cechę f_2, która ponadto nie wykorzystuje żadnych operatorów selekcji (por. tabela 7.3). Jej interpretacja jest następująca: do klasy decyzyjnej nr 10 należy klasyfikować obrazy, w których w wyniku segmentacji

obrazu otrzymuje się regiony o średnim poziomie (operator "$a") składowej czerwonej (cecha atomowa "cR") zawierającym się w przedziale dyskretyzacji nr 4, czyli przyjmującym wartości w zakresie <0.55,+∞)[5] (por. tabela 7.4).

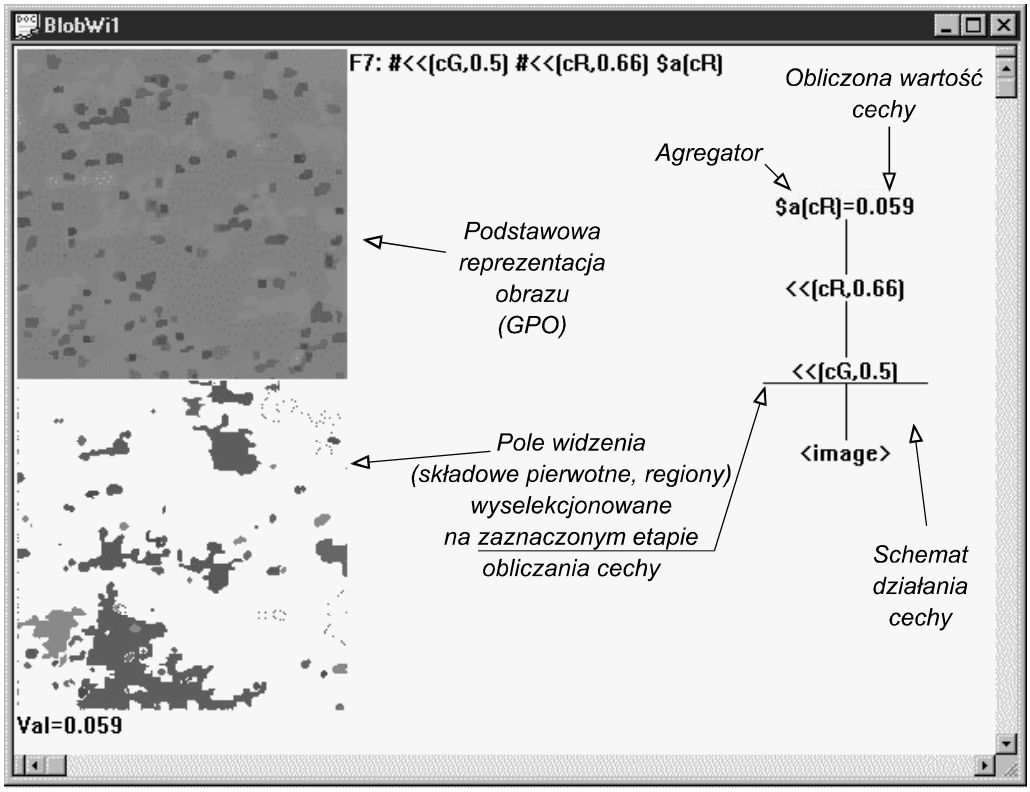
Z kolei interpretacja reguły nr 4 jest już nieco bardziej skomplikowana. Co prawda opiera się ona także tylko na jednej przesłance, ale wykorzystywana przez nią cecha f_7 zawiera dwa operatory selekcji (tabela 7.3). Mówi ona, iż do klasy decyzyjnej nr 10 należy klasyfikować obiekty, dla których wartość cechy f_7 plasuje się w przedziale dyskretyzacji nr 3, tj. przyjmuje wartości z przedziału <0.55, +∞). Cecha f_7 to, podobnie jak w przypadku reguły nr 3, średni poziom składowej czerwonej ("$a(cR)"), obliczony jednak dla odpowiednio wyselekcjonowanych regionów, tj. takich, których składowa zielona nie przekracza wartości 0.5 ("#<(cG,0.5)"), a składowa czerwona nie przekracza wartości 0.66 ("#<(cR,0.66)").

Jak już prezentowano przy omawianiu poprzedniego zastosowania, implementacja komputerowa proponowanego podejścia umożliwia śledzenie procesu obliczania cech, a w szczególności sposobu przeprowadzania przez nie selekcji składowych pierwotnych. Dotyczy to także opisywanego tu zastosowania. Rysunki 7.3 i 7.4 prezentują proces obliczania cechy f_7 ("#<(cG,0.5) #<(cR,0.66) $a(cR)") dla wybranego obrazu reprezentującego klasę decyzyjną nr 9. W lewym górnym narożniku okienka prezentowana jest reprezentacja podstawowa obrazu (obraz oryginalny poddany segmentacji segmentacji), poniżej zaś widoczne są wyselekcjonowane regiony, odpowiadające etapowi przetwarzania podkreślonemu w grafie ilustrującym cechę w prawej części okna. Na rysunku 7.3 widoczne jest pole widzenia otrzymane po zastosowaniu operatora selekcji #<(cG,0.5), zaś rysunek 7.4 zawiera pole widzenia po przeprowadzeniu kolejnego etapu przetwarzania, tj. zastosowaniu operatora selekcji #<(cR,0.66). Przy porównywaniu rysunków 7.3 i 7.4 widoczny jest proces stopniowego zawężania pola widzenia.

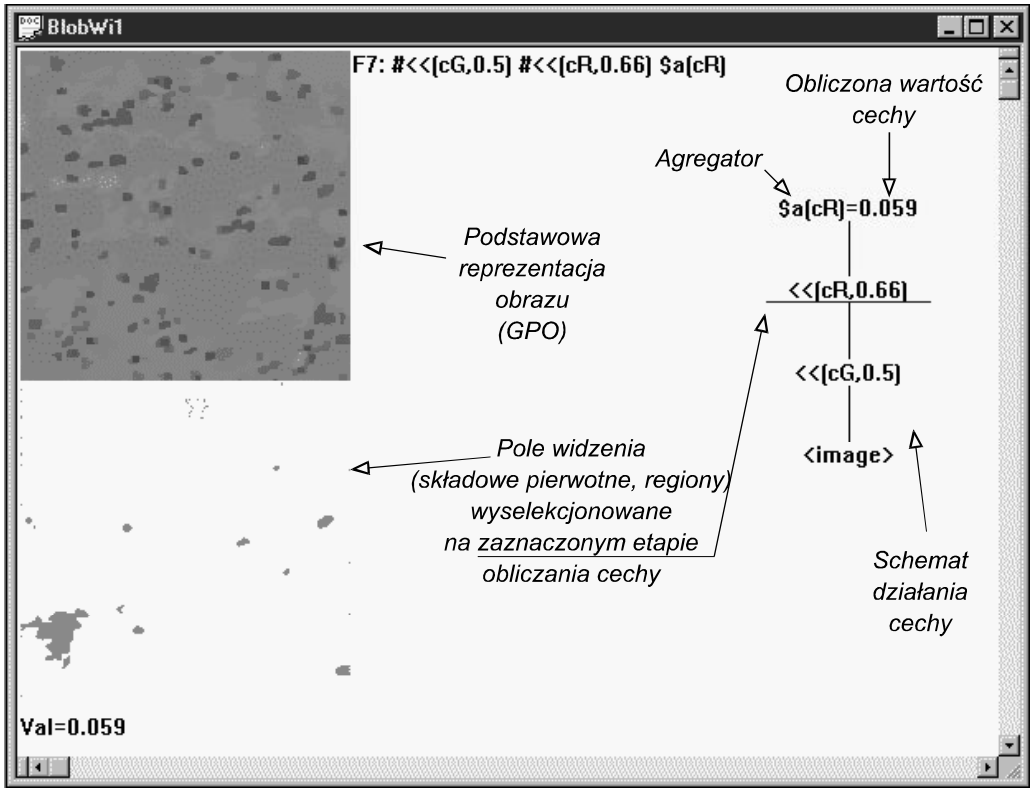
Na zrzutach ekranów implementacji komputerowej widać także, iż wartość cechy f_7 dla prezentowanego przykładu/obrazu wynosi 0.059, co według tabeli 7.4 mieści się w pierwszym przedziale dyskretyzacji (−∞,0.32) i w konsekwencji odpowiada wartości dyskretnej (kodowi) 0. Wiedza pozyskana w procesie konstruktywnej indukcji cech potwierdza się zatem w odniesieniu do tego obrazu, ponieważ $f_7 = 0$ blokuje spełnienie przesłanki reguły nr 4 (por. tabela 7.5), wskazującej na inną klasę decyzyjną, natomiast potencjalnie umożliwia "odpalenie" (ang. *firing*) reguły nr 2, wskazującej na klasę decyzyjną nr 9, do której należy analizowany obraz.

Poza dobrą zdolnością wyjaśniania decyzji, reguły prezentowane w ramach niniejszego przykładu charakteryzują się także **wysoką zdolnością predykcyjną**. Spośród 280 obiektów ze zbioru testującego T (obejmującego wszystkie 14 klas de-

[5] Wartości wszystkich cech atomowych są znormalizowane do przedziału $\langle 0, 1\rangle$.



Rysunek 7.3: Ilustracja procesu obliczania wartości cechy (w szczególności selekcji pola widzenia) w implementacji komputerowej (rysunek prezentuje pole widzenia po zastosowaniu **pierwszego selektora** cechy f_7 omawianej w tekście)



Rysunek 7.4: Ilustracja procesu obliczania wartości cechy (w szczególności selekcji pola widzenia) w implementacji komputerowej (rysunek prezentuje pole widzenia po zastosowaniu **drugiego selektora** cechy f_7 omawianej w tekście)

cyzyjnych, w tym $2 \times 20 = 40$ obiektów z interesujących nas klas decyzyjnych 9 i 10), klasyfikują one 31 obrazów, z czego 25 poprawnie (w pozostałych przypadkach żadna z czterech reguł nie jest "odpalana").

Osłabiając ograniczenia narzucone na generowanie opisu charakterystycznego, można oczywiście otrzymać bardziej obszerny zbiór reguł, pomocny przy wspomaganiu diagnozowania pozostałych klas decyzyjnych. Przykładowo dla **poziomu dyskryminacji 90%i minimalnej siły 5** generowanych jest 216 reguł; tabele 7.6 i 7.7 prezentują część tego zbioru, tj. po trzy najsilniejsze reguły dla poszczególnych klas decyzyjnych. Jak widać reguły pokrywają **wszystkie klasy decyzyjne**, choć ich siła jest zróżnicowana, w zależności od siły wzorców obecnych w informacji obrazowej.

Nr	Reguła	Pokrycie	Wsparcie
1	$(f_0 = 1)\&(f_1 = 0)\&(f_3 = 1)\&(f_4 = 0)\&(f_5 = 2) => (D = 0)$	13	12
2	$(f_1 = 0)\&(f_4 = 0)\&(f_6 = 1)\&(f_7 = 1) => (D = 0)$	7	7
3	$(f_3 = 1)\&(f_4 = 0)\&(f_6 = 1)\&(f_7 = 1) => (D = 0)$	7	7
4	$(f_1 = 1)\&(f_2 = 2)\&(f_3 = 2) => (D = 1)$	11	10
5	$(f_2 = 2)\&(f_3 = 2)\&(f_6 = 3) => (D = 1)$	10	9
6	$(f_0 = 0)\&(f_2 = 2)\&(f_4 = 0) => (D = 1)$	8	8
7	$(f_0 = 0)\&(f_1 = 0)\&(f_2 = 1)\&(f_5 = 3) => (D = 2)$	5	5
8	$(f_0 = 0)\&(f_2 = 3)\&(f_4 = 3) => (D = 2)$	4	4
9	$(f_1 = 0)\&(f_2 = 1)\&(f_3 = 1)\&(f_5 = 3) => (D = 2)$	4	4
10	$(f_1 = 2)\&(f_2 = 2) => (D = 3)$	8	8
11	$(f_0 = 1)\&(f_1 = 1)\&(f_4 = 2) => (D = 3)$	8	8
12	$(f_0 = 1)\&(f_1 = 1)\&(f_6 = 1)\&(f_7 = 1) => (D = 3)$	8	8
13	$(f_1 = 3)\&(f_4 = 1) => (D = 4)$	6	6
14	$(f_1 = 2)\&(f_4 = 2)\&(f_5 = 1) => (D = 4)$	4	4
15	$(f_0 = 1)\&(f_1 = 2)\&(f_4 = 2) => (D = 4)$	3	3
16	$(f_0 = 1)\&(f_2 = 1)\&(f_5 = 3)\&(f_7 = 1) => (D = 5)$	11	11
17	$(f_0 = 1)\&(f_2 = 1)\&(f_5 = 3)\&(f_6 = 2) => (D = 5)$	8	8
18	$(f_0 = 1)\&(f_1 = 1)\&(f_2 = 1)\&(f_6 = 2) => (D = 5)$	7	7
19	$(f_0 = 1)\&(f_1 = 3)\&(f_3 = 0)\&(f_5 = 2) => (D = 6)$	4	4
20	$(f_0 = 1)\&(f_3 = 0)\&(f_4 = 3)\&(f_5 = 2) => (D = 6)$	4	4
21	$(f_1 = 3)\&(f_3 = 0)\&(f_4 = 3)\&(f_5 = 2) => (D = 6)$	4	4
22	$(f_0 = 0)\&(f_2 = 1)\&(f_6 = 3) => (D = 7)$	6	6
23	$(f_1 = 1)\&(f_2 = 1)\&(f_3 = 1)\&(f_6 = 3) => (D = 7)$	6	6
24	$(f_3 = 3)\&(f_5 = 2) => (D = 7)$	5	5
25	$(f_4 = 3)\&(f_5 = 1)\&(f_7 = 1) => (D = 8)$	5	5
26	$(f_0 = 0)\&(f_1 = 1)\&(f_2 = 1)\&(f_4 = 1)\&(f_6 = 1) => (D = 8)$	5	5
27	$(f_0 = 0)\&(f_1 = 1)\&(f_2 = 1)\&(f_5 = 2)\&(f_6 = 1) => (D = 8)$	5	5
28	$(f_2 = 0)\&(f_6 = 3) => (D = 9)$	15	15
29	$(f_6 = 3)\&(f_7 = 0) => (D = 9)$	15	15
30	$(f_1 = 2)\&(f_2 = 0) => (D = 9)$	12	12

(ciąg dalszy na następnej stronie)

Tabela 7.6: Zbiór najsilniejszych reguł wyindukowanych dla reprezentacji z tabeli 7.3, przy poziomie dyskryminacji 90% i minimalnej sile reguły 5 (D – atrybut decyzyjny)

Nr	Reguła	Pokrycie	Wsparcie
31	$(f_7 = 3) => (D = 10)$	25	24
32	$(f_2 = 4) => (D = 10)$	24	23
33	$(f_2 = 3)\&(f_6 = 2) => (D = 10)$	6	6
34	$(f_4 = 1)\&(f_5 = 1)\&(f_6 = 1) => (D = 11)$	14	13
35	$(f_3 = 1)\&(f_5 = 1)\&(f_7 = 1) => (D = 11)$	13	12
36	$(f_5 = 1)\&(f_6 = 1)\&(f_7 = 1) => (D = 11)$	13	12
37	$(f_0 = 1)\&(f_2 = 3)\&(f_4 = 0) => (D = 12)$	15	14
38	$(f_0 = 1)\&(f_4 = 0)\&(f_7 = 2) => (D = 12)$	14	14
39	$(f_4 = 0)\&(f_6 = 0)\&(f_7 = 2) => (D = 12)$	13	12
40	$(f_0 = 1)\&(f_1 = 3)\&(f_4 = 2) => (D = 13)$	6	6
41	$(f_1 = 3)\&(f_4 = 2)\&(f_5 = 0) => (D = 13)$	6	6
42	$(f_1 = 3)\&(f_3 = 1) => (D = 13)$	4	4

Tabela 7.7: Ciąg dalszy tabeli 7.6

Rozdział 8

Wnioski końcowe i kierunki dalszych badań

8.1 Podsumowanie wyników pracy

Niniejsza rozprawa prezentuje nowe podejście do automatycznego poszukiwania suboptymalnych reprezentacji obrazu (konstruktywnej indukcji cech obrazu) dla potrzeb wnioskowania na podstawie informacji obrazowej, a w szczególności rozpoznawania obrazów. W tym celu wykorzystuje ono po części dorobek rozpoznawania obrazów, wspomagania decyzji i uczenia maszynowego.

Prowadzone prace doprowadziły do uzyskania wyników, które można podzielić na **metodyczne, empiryczne i informatyczne**. Do najistotniejszych z uzyskanych wyników metodycznych należą:

- zaproponowanie **uniwersalnego sposobu integracji konstruktywnej indukcji cech i rozpoznawania obrazów**, poprzez wykorzystanie pewnej odpowiedniości pomiędzy operatorami KI a etapami wnioskowania z informacji obrazowej, w tym koncepcji:
  - podstawowej reprezentacji obrazu,
  - dekompozycji przetwarzania realizowanego w ramach cechy na **ciąg operatorów**,
- zaproponowanie jednorodnej **formalizacji (unifikacji) problemów selekcji cech i konstruktywnej indukcji cech**,

- zaproponowanie zmodyfikowanego sposobu przeprowadzania eksperymentu wielokrotnego uczenia i testowania w modelu *wrapper* (**odwrotna walidacja skrośna**).

Daleko posunięta **integracja uczenia z procesem przetwarzania i rozpoznawania obrazu** daje w rezultacie podejście charakteryzujące się, potwierdzonymi przez weryfikację empiryczną, zaletami:

- **mniejszą kosztownością i czasochłonnością projektowania systemu WDIO**, jako konsekwencją zastąpienia w znacznej mierze eksperta dziedziny zastosowania w procesie doboru metod przetwarzania obrazu i ekstrakcji cech,

- tworzeniem **zwięzłych reprezentacji obrazu** zapewniających stosunkowo dobrą trafność klasyfikowania (rozpoznawania), co jest szczególnie istotne w rozpoznawaniu obrazów, gdzie przekształcenie obrazu stanowiącego dużą ilość informacji do odpowiedniej i zwartej reprezentacji stanowi główną trudność. Eksperymenty pokazały także, iż **proces konstruktywnej indukcji cech wymusza dywersyfikację cech** (por. rozdział 5), co objawia się m.in. wyznaczaniem wartości poszczególnych cech na podstawie różnych fragmentów obrazu i, w konsekwencji, znaczną odpornością na wzorce zaszumione i/lub niekompletne,

- **większą uniwersalnością w porównaniu z konwencjonalnymi podejściami dwuetapowymi**. Dzięki uczeniu ten sam system (lub poddany niewielkim modyfikacjom) może być wykorzystywany do uczenia się i poszukiwania cech dyskryminujących w pokrewnych zastosowaniach. Co więcej, dostosowanie reprezentacji podstawowej i operatorów KI umożliwia stosowanie podejścia w odległych zastosowaniach, jak to pokazały eksperymenty opisywane w rozdziałach 6 i 7. Choć oczywiście przenoszalność doświadczenia na inne zastosowania jest ograniczona, to jest to ograniczenie dużo mniejsze niż w przypadku podejść konwencjonalnych.

Jak pokazano w punkcie 6.3.4, **interpretacja otrzymywanych cech** f (i, w konsekwencji, całych reprezentacji F) jest stosunkowo czytelna, w odróżnieniu od cech używanych w podejściach konwencjonalnych, które często mają charakter "techniczny". Jest to szczególnie istotne, gdy na przykład chcemy przekonać do wykorzystywanych cech (lub całego systemu wnioskującego) eksperta dziedziny zastosowania. Tę cechę proponowane rozwiązanie zawdzięcza m.in. **reprezentowaniu procesu wnioskowania z informacji obrazowej na pośrednim poziomie abstrakcji** (w postaci selektorów i agregatorów), w odróżnieniu od większości rozwiązań wykorzystujących uczenie prezentowanych w literaturze, gdzie operuje się

bezpośrednio na mapach bitowych lub, wręcz przeciwnie, na globalnym przetwarzaniu i analizie obrazu (por. rozdział 3.3). Przez analogię można tu zatem mówić o realizowaniu przez proponowane podejście swoistej **eksploracji danych i odkrywania wiedzy** (ang. *data mining, knowledge discovery*) **w odniesieniu do danych obrazowych** (por. [Fayyad, Piatetsky–Shapiro, *et al.* 1996], [Michalski, Bratko, *et al.* 1997]).

Pewną zaletą proponowanego podejścia jest także jego **modułowa konstrukcja**, która ułatwia wykorzystywanie różnych (meta)heurystyk do przeszukiwania przestrzeni reprezentacji i różnych funkcji oceniających E. Ponadto wydaje się, że proponowane podejście dobrze odpowiada podstawowym **założeniom komputerowego wspomagania decyzji**, gdzie nie dąży się do wyeliminowania eksperta, a jedynie oferuje mu metody i narzędzia wyrażania preferencji i reprezentacji wiedzy. Wkład eksperta obejmuje wybór odpowiedniej reprezentacji podstawowej obrazu $r(x)$, zaprojektowanie operatorów selekcji S i agregacji A oraz dobór parametrów algorytmu przeszukiwania przestrzeni reprezentacji F, natomiast **podejście realizuje poszukiwanie suboptymalnego programu przetwarzania i analizy obrazu w zakresie od podstawowej reprezentacji obrazu do zwięzłego opisu** F^+ wyrażonego w kategoriach wyindukowanych cech. Wydaje się, że jest to znaczne uproszczenie w stosunku do sytuacji konwencjonalnych podejść dwuetapowych, gdzie projektant musi samodzielnie tworzyć różne reprezentacje obrazu i weryfikować ich przydatność w kontekście zastosowania.

Do **wyników informatycznych** należy zaliczyć przeprowadzenie analizy złożoności obliczeniowej proponowanego podejścia i jego implementację, w której wprowadzono wiele usprawnień mających na celu przyspieszenie obliczeń (dodatek B).

8.2 Możliwości udoskonaleń podejścia i dalszych badań

Prace prowadzone nad rozprawą zaowocowały pewnymi spostrzeżeniami i pomysłami, które mogą prowadzić do dalszych udoskonaleń metody i kontynuacji badań w nurcie konstruktywnej indukcji cech obrazu. Poniżej zaprezentowanych jest kilka z nich.

1. W omawianym podejściu jedyną wielkością sterującą procesem przeszukiwania przestrzeni reprezentacji F jest funkcja oceniająca E. Nasuwa się pytanie, czy sama wartość funkcji E dla zbioru cech powinna wyznaczać kierunki przeszukiwania przestrzeni F ? Wydaje się, że można by zdefiniować inne czynniki, które pomogłyby w zawężeniu zbioru rozważanych (tj. ocenianych przy pomocy E) rozwiązań, eliminując te, o których z góry wiadomo, iż z dużym prawdopodobieństwem nie

dadzą rozwiązań lepszych niż bieżące. Prowadziłoby to w konsekwencji do zmniejszenia liczności sąsiedztwa N przy lokalnym przeszukiwaniu przestrzeni rozwiązań i przyspieszenia obliczeń, co dałoby możliwość uzyskiwania lepszych rezultatów w tym samym czasie i/lub rozwiązywania problemów o większych rozmiarach.

Ideę tę w pewnym stopniu realizuje wstępna selekcja (pojedynczych) cech opisywana w dodatku B, której wykorzystanie rzeczywiście prowadzi do znacznego usprawnienia obliczeń bez znaczącego pogorszenia otrzymywanych wyników. Można by jednak pójść dalej i **wyposażyć system konstruktywnej indukcji w dodatkową wiedzę**, która na przykład pomagałaby decydować, czy dla danego zbioru (obrazów) uczącego L w ogóle warto rozważać wykorzystanie jakiegoś operatora selekcji pola widzenia. Omawiane podejście zyskało by w ten sposób cechy konstruktywnej indukcji cech sterowanej wiedzą (KCI, por. punkt 4.4). Problem ten wiąże się w pewnym stopniu z zagadnieniem automatycznego pozyskiwania przez system uczący się **metawiedzy** (doświadczenia), która może być wykorzystana do rozwiązywania innych (podobnych) zadań (por. np. [Bensusan 1998]).

2. Proponowane podejście zakłada "liniowy" charakter przetwarzania, gdzie cecha f utożsamiana jest z *ciągiem* operatorów $f^{(i)}$, a konstruktywny operand jest zawsze pojedynczym polem widzenia $R(x)$. Naturalnym uogólnieniem tej koncepcji jest uwzględnienie operatorów KI, które:

- **obejmują wiele pól widzenia** $R^{(i)}(x)$ (np. selektor typu *suma*, tworzący sumę teoriomnogościową dwóch lub więcej pól widzenia),
- poza polem widzenia $R(x)$ akceptują **argumenty innych typów** (np. wielkości skalarne obliczone przez inną cechę ("podcechę"); jest to swojego rodzaju "zagnieżdżanie" cech),
- przeprowadzają obliczenia na wielkościach skalarnych (liczboweych) wyznaczonych przez inne cechy (czyli charakterystyczne dla konwencjonalnej konstruktywnej indukcji, por. przykład 4).

Tak uogólnione operatory prowadzą do interpretacji graficznej cechy w postaci grafu skierowanego o jednym wierzchołku początkowym (ang. *source*), którym jest podstawowa reprezentacja obrazu $r(x)$, oraz jednym wierzchołku końcowym (ang. *sink*), odpowiadającym obliczonej wartości cechy. Zbiór cech możliwych do rozważenia w ramach tak poszerzonej reprezentacji jest oczywiście daleko szerszy od zbioru cech reprezentowanych przez ciągi operatorów. Obecna wersja implementacji komputerowej proponowanego podejścia akceptuje już taką składnię cech.

3. Konstruktywna indukcja cech w omawianym podejściu ograniczona jest z jednej strony podstawową reprezentacją obrazu $r(x)$, z drugiej zaś skalarnym charakterem wartości zwracanych przez poszczególne cechy $f_j \in F^+$. Interesujące wydaje

się rozszerzenie tego procesu w obu tak rozumianych kierunkach. W szczególności, dość naturalnym rozwiązaniem byłoby **włączenie do procesu KI wstępnego przetwarzania prowadzącego do wyznaczenia podstawowej reprezentacji obrazu**. W ten sposób proces konstruktywnej indukcji mógłby zyskać nowe "stopnie swobody", polegające np. na możliwości wykonywania ruchów modyfikujących sposób (lub parametry) obliczania podstawowej reprezentacji obrazu. Z drugiej strony, indukcję cech można by powiązać jeszcze ściślej z generowaniem klasyfikatora; dla pewnych typów algorytmów indukcji (np. drzew decyzyjnych) da się to zrobić w bardzo elegancki sposób.

4. Ciekawe rozwinięcie proponowanego podejścia mogłoby dotyczyć funkcji oceniających E uwzględniających specyfikę procesu konstruktywnej indukcji cech w **wieloklasowych problemach uczenia maszynowego**. Przesłanką jest tu spostrzeżenie, iż w przypadku stopniowego (przyrostowego) sposobu działania algorytmu tworzącego reprezentację obrazu F, trafność klasyfikowania wykorzystywana w metodzie *wrapper* nie zawsze dobrze ocenia poszczególne rozwiązania (np. w sąsiedztwie). Innymi słowy, łatwo jest podać przykład, w którym dla dwóch reprezentacji F' i F'' zajdzie $E(F') = E(F'')$, a będą jednocześnie istniały inne przesłanki za preferowaniem F' lub F'', wynikające z oceny trafności klasyfikowania dla poszczególnych *par* klas (widocznej np. w tzw. macierzy pomyłek (ang. *confusion matrix*)).

Ponadto warto też zwrócić uwagę na fakt, iż, poza wspomaganiem decyzji z informacji obrazowej, proponowane podejście nadaje się do wnioskowania z (analizowania, rozpoznawania, etc.) dużo szerszej klasy danych. Patrząc na nie szerzej, jest to pewna **metodyka konstrukcji cech na podstawie przykładów opisanych w sposób złożony,** w szczególności strukturalny (w odróżnieniu od konwencjonalnej indukcji konstruktywnej, gdzie podstawą jest pewien wektor (zbiór) cech). Spostrzeżenie to znacząco poszerza zakres potencjalnych zastosowań podejścia (np. na rozpoznawanie sygnałów/przebiegów czasowych).

Dodatek A

Zestawy operatorów KI

W eksperymentach obliczeniowych opisywanych w rozdziale 6 wykorzystywano różne zestawy cech podstawowych F_0 i operatorów KI O. Zgodnie z uwagą zawartą w punkcie 5.2.3, ze względów praktycznych ruchy nie są definiowane *explicite*, lecz poprzez znacznie bardziej zwięzłe *szablony*, które można traktować jako uproszczone gramatyki. W zapisie szablonów, przechowywanych jako pliki tekstowe, przyjęto następujące konwencje:

- agregatory poprzedzane są znakiem '$',
- selektory poprzedzane są znakiem '#',
- zapis '$v = lista$' oznacza, że w dalszej części pliku opisującego szablon zmienna (symbol nieterminalny w terminologii gramatyk formalnych) v ma być zastępowany dowolną wartością (symbolem nieterminalnym) z listy *lista* umieszczonej po prawej stronie znaku '=',
- zapis '$+n$' oznacza zmienną (symbol nieterminalny), który ma być zastępowany dowolną (wcześniej wygenerowaną z szablonu) cechą o długości co najwyżej n.

Poniższa tabela A.1 prezentuje szablony zestawów użytych w eksperymentach dotyczących **rozpoznawania znaków** (rozdział 6) oraz ich liczności. Interpretacja pozostałych symboli (nazw cech atomowych, selektorów i agregatorów) podana jest w rozdziale 6.

W eksperymentach obliczeniowych dotyczących zastosowania w **diagnostyce medycznej** (rozdział 7) stosowano jeden zestaw cech podstawowych F_0 i operatorów KI O, zaprezentowany w tabeli A.2.

Zestaw	*Szablon zbioru* F_0	*Szablon zbioru* O	$\|F_0\|$	$\|O\|$

Z_A	a=x y d n
	$%a
	$M(%a)
	$m(%a)
	$a(%a)
	$d(%a)
		#a([0,1,2,3])
		c=0.0 0.25 0.5 0.75 1.0
		q=0.1 0.2 0.3 0.4 0.5
		#r%q(%c,%c)
			20	129

Z_B	jak wyżej, oraz:
	#a([0,1,2,3]) +1
	c=0.0 0.25 0.5 0.75 1.0
	q=0.1 0.2 0.3 0.4 0.5
	#r%q(%c,%c) +1
		jak wyżej	2600	129

Z_C	jak w Z_A, przy czym a=x y d n X Y eX eY A	jak w Z_A, oraz #?d(%a) #?m(%a) #?M(%a) #?<(%a,$%a) #?<(%a,[0,0.1,0.25,0.33, 0.5,0.66,0.75,0.9,1])	45	257

Tabela A.1: Zestawy operatorów wykorzystywane w rozpoznawaniu znaków

Szablon zbioru F_0	*Szablon zbioru* O	$\|F_0\|$	$\|O\|$
f=A cR cG cB	f=A cR cG cB	20	128
$%f	#?d(%f)		
$M(%f)	#?m(%f)		
$m(%f)	#?M(%f)		
$a(%f)	#?<(%f,$%f)		
$d(%f)	#?<(%f,[0,0.1,0.25,0.33, 0.5,0.66,0.75,0.9,1])		

Tabela A.2: Zestawy operatorów wykorzystywane w zastosowaniu medycznym

Dodatek B

Implementacja systemu KI cech obrazu

Sposoby usprawnienia obliczeń

W implementacji podejścia zastosowano wiele usprawnień mających na celu przyspieszenie obliczeń. Większość z nich charakter czysto "techniczny", tj. nie wpływa na wynik; są jednak także i takie, które prowadzą do pominięcia pewnych zbiorów cech podczas przeglądania przestrzeni rozwiązań. Ze zrozumiałych względów dotyczyły one głównie tych części algorytmu, które były najbardziej czasochłonne. Wyznaczenia tych fragmentów dokonano na podstawie analizy algorytmu (szacowanie złożoności obliczeniowej dla poszczególnych fragmentów podejścia) i eksperymentalnych pomiarów implementacji (badanie krytycznych obliczeniowo wezłów programu, *profiling*).

Sąsiedztwo $N(F)$ bieżącego stanu F w algorytmie stromego przeszukiwania lokalnego SLS może być bardzo liczne. Zgodnie z wywodem przeprowadzonym w punkcie 5.4, pesymistyczne oszacowanie rozmiaru sąsiedztwa $N(F)$, czyli liczby stanów ocenianych przy wykonywaniu pojedynczego ruchu algorytmu SLS to (por. wzór 5.7):

$$|F_0| + |F|_{\max} \left(2\,|O|\,|f|_{\max} + |f|_{\max} + 1\right).$$

Nawet dla najmniejszego ze stosowanych w przeprowadzanych eksperymentach zestawu operatorów (Z_A) jest to kilka tysięcy stanów (rozwiązań). Przeglądanie tak licznego sąsiedztwa w każdym ze stanów odwiedzanych podczas działania algorytmu SLS byłoby bardzo kosztowne pod względem obliczeniowym, stąd niezbędne jest wprowadzenie dodatkowych mechanizmów ograniczających rozmiar $N(F)$.

W implementacji podejścia zdecydowano się zatem wprowadzić dodatkową fazę polegającą na *ocenianiu i wstępnej selekcji pojedynczych cech* powstałych przez mo-

dyfikowanie bieżącego rozwiązania (por. punkt 5.2.4). W tym celu wprowadzono funkcję (miarę) e służącą do oceny pojedynczych cech. Zdefiniowano ją w oparciu o entropię (ang. *[mutual] entropy*), a dokładniej zysk na informacji (ang. *info gain*), jedną z najbardziej popularnych miar skuteczności dyskryminacyjnej atrybutów w uczeniu maszynowym, wykorzystywaną m.in. powszechnie w algorytmach indukcji drzew decyzyjnych, np. C4.5 [Quinlan 1992]. Poza tym za wyborem tej miary przemawiały:

- stosunkowo niska złożoność obliczeniowa,
- zdolność do oceniania atrybutów ciągłych,
- odporność na szumy i "odstające" wartości cech.

Entropię zbioru przykładów X dla nadzorowanego dwuklasowego[1] problemu uczenia maszynowego definiujemy następująco:

$$H(X) = -p^+(X)\log_2 p^+(X) - p^-(X)\log_2 p^-(X) \tag{B.1}$$

gdzie prawdopodobieństwa p^+ i p^- obliczane są według wzorów:

$$p^+(X) = \frac{|X^+|}{|X|}, \quad p^-(X) = 1 - p^+(X) = \frac{|X^-|}{|X|}$$

Najważniejsze własności entropii to:

1. $\forall X \subseteq U \quad H(X) \in \langle 0, 1\rangle$
2. $X^+ = \emptyset \vee X^- = \emptyset \Rightarrow H(X) = 0$
3. $|X^+| = |X^-| \Rightarrow H(X) = 1$

Funkcja e działa niemal dokładnie w taki sposób, w jaki oceniany jest atrybut ciągły i ustalany jest na nim punkt cięcia w popularnym algorytmie generowania drzew decyzyjnych C4.5 [Quinlan 1992] czy w znanej metodzie dyskretyzacji atrybutów ciągłych Fayyada-Iraniego [Fayyad & Irani 1992]. Przykłady x ze zbioru X są najpierw porządkowane według rosnących wartości $f(x)$ przyjmowanych przez ocenianą cechę f dla zbioru uczącego. Następnie rozważane są wszystkie możliwe podziały zbioru X na pary rozłącznych podzbiorów $(X_L(c), X_R(c))$, wyznaczone

[1] Dla uproszczenia rozważamy problem dwuklasowy; uogólnienie na problem wieloklasowy jest proste.

przez wartość $c \in D^{-1}(f)$ przypadającą pomiędzy dwoma kolejnymi różnymi wartościami pochodzącymi ze zbioru uczącego.

$$X_L(c) = \{x \in X : f(x) < c\}, \quad X_R(c) = \{x \in X : f(x) \geq c\}$$

Dalej wybierany jest ten podział, który daje największy *zysk na entropii*, zdefiniowany jako:

$$G(X, c) = H(X) - H(X|c)$$

gdzie:

$$H(X|c) = \frac{|X_L(c)|}{|X|} H(X_L(c)) + \frac{|X_R(c)|}{|X|} H(X_R(c))$$

Ostatecznie:

$$e(f) = \max_{c \in C_{cut}} G(X, c)$$

gdzie C_{cut} jest zbiorem rozważanych punktów cięcia. Ze względów obliczeniowych dąży się oczywiście do ograniczenia liczby rozważanych punktów cięcia. W najbardziej ogólnym przypadku C_{cut} jest minimalnym zbiorem wartości c indukujących wszystkie unikalne podziały $(X_L(c), X_R(c))$ zdefiniowane jak wyżej; taką własność zapewnia np. $C_{cut} = \{f(x) : x \in X\}$ (por. np. C4.5 [Quinlan 1992]).

W proponowanym podejściu wprowadzono dodatkowe usprawnienie, polegające na zawężeniu zbioru rozważanych punktów cięć C_{cut}. Dla dwóch klas decyzyjnych C_1, C_2, równolicznych ($|C_1| = |C_2|$), lub w przybliżeniu równolicznych, z góry wiadomo, że podziały (X_L, X_R) silnie "niezrównoważone", tj. takie, że $|X_L| \ll |X_R|$ lub $|X_L| \gg |X_R|$, nie mogą dać wysokiej wartości funkcji e, gdyż w takiej sytuacji jeden z podzbiorów X_L, X_R musi zawierać stosunkowo dużo przykładów z obu klas decyzyjnych, co pogarsza (zwiększa) wartość entropii H. W implementacji podejścia nie rozważa się zatem takich punktów cięć c, które prowadzą do podziałów (X_L, X_R) takich, że $\min(|X_L|, |X_R|) < c_{\min}$, gdzie $c_{\min}$ jest parametrem ustalanym przez użytkownika.

Zysk na entropii e wprowadza oczywiście do proponowanego podejścia pewne **dodaktowe ukierunkowanie indukcyjne**, polegające na preferowaniu cech grupujących przykłady pozytywne i negatywne w dwóch (i tylko dwóch) różnych skupieniach. Ujmując to inaczej, zysk na entropii wysoko ocenia tylko te cechy, przy użyciu których da się dobrze dyskryminować przykłady z X za pomocą *pojedynczych* warunków elementarnych typu $>$ i $<$. W związku z tym dla pewnych cech miara e może dać niską ocenę mimo ich znacznej zdolności dyskryminacyjnej, gdy dyskryminowanie wymaga wyrażeń bardziej wyrafinowanych niż pojedyncze warunki $>$ i $<$. Jest to jednak przypadek rzadki; w praktyce przydatność zysku na entropii w poszukiwaniu dobrych cech jest bardzo wysoka. Ponadto proces konstruktywnej

indukcji zapewnia tak wiele stopni swobody, iż wydaje się, że szansa wyindukowania cech dyskryminujących w wyżej opisany sposób jest znaczna (por. rozdział 5). ■

W implementacji komputerowej wprowadzono także wiele innych usprawnień o charakterze czysto technicznym, których pełen opis wykracza poza zakres niniejszej pracy. Jednym z istotniejszych z nich jest przyśpieszenie obliczania wartości funkcji e, co jest stosunkowo czasochłonnym procesem ze względu na obecność funkcji $\log_2$. W tym celu zastosowano tablicę adresowaną zawartością (ang. *lookup table*) do przechowywania wartości wyrażenia pojawiającego się we wzorze B.1, tj. $-p \log_2 p$, dla wartości prawdopodobieństwa $p \in \langle 0, 1 \rangle$ podawanej z dokładnością 0.001. Pomiary eksperymentalne pokazały, że w przypadku średnim zabieg ten skrócił czas znajdowania punktu cięcia według wyżej opisanego algorytmu o ok. 30%.

Obsługa wyjątków

Jak nadmieniono w punkcie 5.2.2, przy obliczaniu wartości agregatora zastosowanego do danego pola widzenia należy liczyć się z koniecznością obsługi pewnych sytuacji nietypowych (wyjątków). Najistotniejszym i najczęściej występującym wyjątkiem jest sytuacja, gdy agregator stosowany jest do *pustego pola widzenia* ($R(x) = \emptyset$). Powstaje wówczas problem: jaką wartość ma zwrócić agregator, jeżeli brak jest podstaw (wartości cech atomowych składowych pierwotnych) do obliczeń?[2] Problem ten wiąże się ze znanym w uczeniu maszynowym zagadnieniem interpretacji i obsługi *brakujących wartości*.

Najprostszym rozwiązaniem, stosowanym niekiedy w prostych technikach uczenia maszynowego, jest wyeliminowanie przykładów i/lub cech dla których pojawiają się brakujące wartości. Jest to jednak rozwiązanie bardzo drastyczne, w jego konsekwencji tracimy bowiem odpowiednio część zbioru uczącego lub część opisu. Ponadto, w konstruktywnej indukcji cech obrazu brakująca wartość powstała wskutek pustego pola widzenia nie musi być wcale (i z reguły nie jest) sytuacją nieprawidłową, lecz wręcz przeciwnie: pewną regularnością w zbiorze uczącym, którą można wykorzystać do dyskryminowania klas decyzyjnych.

Dlatego w proponowanym podejściu nie rezygnuje się z cech z brakującymi wartościami powstałymi wskutek pustego pola widzenia i przyjmuje w takiej sytuacji $f(x) = \phi$. Jednak symbol ϕ oznaczający wartość brakującą nastręcza problemów, gdy trzeba obliczyć wartość $e(f)$ funkcji oceniającej pojedynczą cechę (por. poprzedni punkt). W związku z tym korzysta się z faktu, iż funkcja e bazuje na poszukiwaniu optymalnego punktu cięcia i powstałego w ten sposób podziału zbioru

[2] Problem ten nie dotyczy niektórych operatorów, w tym na przykład agregatora $num[R(x)]$ (por. punkt 6.2.2), obliczającego ilość składowych pierwotnych w $R(x)$.

przykładów X na dwa rozłączne podzbiory (X_L, X_R). Jeżeli w trakcie obliczania cechy dla przykładów z X wystąpi puste pole widzenia, to ocenia się cechę dwukrotnie, przyjmując, że brakujące wartości przypadają na lewo lub na prawo od punktu cięcia (przyjmując np. raz $\phi = \lfloor D^{-1}(f) \rfloor$, a za drugim razem $\phi = \lceil D^{-1}(f) \rceil$). Następnie z tak uzyskanych ocen wybiera się lepszą. Wyniki eksperymentów obliczeniowych pokazały, że takie "obsługiwanie" wartości brakujących jest warte zachodu: wśród wyindukowanych cech jest zazwyczaj kilka procent takich, które zostały włączone do rozwiązania dzięki rozważeniu podanych wyżej dwóch możliwych interpretacji brakujących wartości.

Bibliografia

[Aha 1991] Aha, D.W. Instance–Based Learning Algorithms. *Machine Learning*, 6, 1991, ss. 37–66.

[Ahuja 1995] Ahuja, N. On Detection and Representation of Multiscale Low–Level Image Structure. *ACM Computing Surveys*, 27(3), 1995.

[Alam & Karim 1998] Alam, M.S., Karim, M.A. Advances in Recognition Techniques, Part 2. *OptEng*, 37(1), 1998, ss. 732–734.

[Aloimonos 1993] Aloimonos, Y.(red.) *Active perception.* Lawrence Erlbaum Associates, Hillsdale, 1993.

[Aloimonos, Fermüller, *et al.* 1995] Aloimonos, Y., Fermüller, C., Rosenfeld, A. Seeing and Understanding: Representing the Visual World. *ACM Computing Surveys*, 27(3), 1995.

[Bellman 1961] Bellman, R.E. *Adaptive Control Processes.* Princeton University Press, 1961.

[Bensusan 1998] Bensusan, H.N. *Automatic bias learning: an inquiry into the inductive basis of induction. Praca doktorska. School of Computing and Cognitive Sciences – University of Sussex.* 1998.

[Biederman 1985] Biederman, I. Human Image Understanding: recent research and a theory. *Computer Vision, Graphics and Image Understanding*, 32(1), 1985.

[Bienenstock & von der Malsburg 1987] Bienenstock, E., von der Malsburg, C. A Neural Network for Invariant Pattern Recognition. *Europhys. Lett.*, 4(1), 1987, ss. 121–126.

[Blakemore 1975] Blakemore, C. *Central visual processing. "Handbook of Psychology".* American Press, New York, 1975.

[Bloedorn & Michalski 1991] Bloedorn, E., Michalski, R.S. Data–driven Constructive Induction in AQ17–PRE. A Method and Experiments. W: *Proceedings of the IEEE International Conference on Tools for AI*, San Jose, CA, 1991, ss. 30–37.

[Błażewicz, Cellary, *et al.* 1983] Błażewicz, J., Cellary, W., Słowiński, R., Węglarz, J. *Badania operacyjne dla informatyków.* Wydawnictwa Naukowo–Techniczne, Warszawa, 1983.

[Bolc & Zaremba 1992] Bolc, L., Zaremba, J. *Wprowadzenie do uczenia się maszyn.* Akademicka Oficyna Wydawnicza, Warszawa, 1992.

[Bouckaert 1988] Bouckaert, A. Medical Diagnosis: Are Expert Systems Needed?. *N Engl J Med*, 350, 1988, ss. 2800–2815.

[Breiman, Friedman, *et al.* 1984] Breiman, L., Friedman, J., Ohlsen, R., Stone, C. *Classification and Regression Trees.* Wadsworth, Monterey, 1984.

[Bubnicki 1990] Bubnicki, Z. *Wstęp do systemów ekspertowych.* PWN, Warszawa, 1990.

[Burr 1988] Burr, D. Experiments on neural net recognition of spoken and written text. *IEEE Trans. on Acoustic, Speech, and Signal Proc.*, 36(7), 1988, ss. 1162–1168.

[Cai & Liu 1999] Cai, J., Liu, Z.Q. Integration of Structural and Statistical Information for Unconstrained Handwritten Numeral Recognition. *IEEE Trans. on PAMI*, 21(3), 1999, ss. 263–270.

[Chan & Grzymała–Busse 1994] Chan, Ch.–Ch., Grzymała–Busse, J.W. On the two local inductive algorithms: PRISM and LEM2. *Foundations of Computing and Decision Sciences*, 19(3), 1994, ss. 185–204.

[Chan & Stolfo 1993] Chan, P.K., Stolfo, S.J. Experiments on multistrategy learning by meta–learning. W: *Proceedings of the Second International Conference on Information and Knowledge Management.* 1993.

[Chan & Stolfo 1993] Chan, P.K., Stolfo, S.J. Meta–learning for Multistrategy and Parallel Learning. W: *Proceedings of the Second International Workshop on Multistartegy Learning.* 1993, ss. 150–165.

[Cholewa & Pedrycz 1987] Cholewa, W., Pedrycz, W. *Systemy doradcze. Politechika śląska. Skrytpy uczelniane Nr 1447.* 1987.

[Cios, Pedrycz, *et al.* 1998] Cios, K.J., Pedrycz, W., Swiniarski, R.W. *Data Mining Methods for Knowledge Discovery.* Kluwer Academic Publishers, Dordrecht/Boston/London, 1998.

[Craven & Shavlik 1994] Craven, M.W., Shavlik, J.W. Using Sampling and Queries to Extract Rules from Trained Neural Networks. W: *Proc. Eleventh Int. Conf. on Machine Learning.* Morgan Kaufmann Publishers, Inc., San Mateo, 1994.

[Dash & Liu 1997] Dash, M., Liu, H. Feature Selection for Classification. *Intelligent Data Analysis*, 1(3), 1997.

[Diederich 1992] Diederich, J. Explanation and artificial neural networks. *Int. J. Man–Machine Studies*, 37, 1992, ss. 335–355.

[Dietterich & Bakiri 1995] Dietterich, T.G., Bakiri, G. Solving multiclass learning problems via error–correcting output nodes. *Journal of Artificial Intelligence Research*, (2), 1995, ss. 263–286.

[Domingos 1997] Domingos, P. Context–sensitive feature selection for lazy learners. *Artificial Intelligence Review*, 11(1–5), 1997, ss. 227–253.

[Draper 1993] Draper, B.A. *Learning Object Recognition Strategies. Praca doktorska. Graduate School of the University of Massachusetts.* 1993.

[Duda & Hart 1972] Duda, R.O., Hart, P.E. Use of the Hough Transform to Detect Lines and Curves in Pictures. *Comm. ACM*, 15(1), 1972, ss. 11–15.

[Duda & Hart 1973] Duda, R.O., Hart, P.E. *Pattern Classification and Scene Analysis.* Wiley, 1973.

[Eshera & Fu 1984] Eshera, M.A., Fu, K.–S. A Graph Distance Measure for Image Analysis. *IEEE Trans. on SMC*, 14(3), 1984.

[Faifer, Janikow, *et al.* 1999] Faifer, M., Janikow, C., Krawiec, K. Extracting fuzzy symbolic representation from artificial neural networks. W: *Proceedings of the 18th International Conference of the North American Fuzzy Information Processing Society. New York.* 1999, ss. 600–604.

[Fawcett, Gordon, *et al.* 1994] Fawcett, T., Gordon, D.F., Sutton, R. *Constructive Induction Needs a Methodology based on Continuing Learning. Panel discussion; ML–COLT'94 Workshop on Constructive Induction and Change of Representation.* 1994.

[Fayyad & Irani 1992] Fayyad, U.M., Irani, K.B. On the Handling of Continuous–Valued Attributes in Decision Tree Generation. *Machine Learning*, (8), 1992, ss. 87–.

[Fayyad, Djorgovski, *et al.* 1996] Fayyad, U.M., Djorgovski, S.G., Weir, N. Automaing the Analysis and Cataloging of Sky Surveys. W: Fayyad, U.M., Piatetsky–Shapiro, G., Smyth, P., Uthurusamy, R.(red.) *Advances in Knowledge Discovery and Data Mining.* AAAI Press / The MIT Press, Cambridge Mass., 1996, ss. 471–493.

[Fayyad, Piatetsky–Shapiro, *et al.* 1996] Fayyad, U.M., Piatetsky–Shapiro, G., Smyth, P., Uthurusamy, R.(red.) *Advances in Knowledge Discovery and Data Mining.* AAAI Press / The MIT Press, Cambridge Mass., 1996.

[Feldman & Bruckstein 1991] Feldman, Y.A., Bruckstein, A.(red.) *Artificial intelligence and computer vision.* Elsevier Science Publishers, Amsterdam, 1991.

[Fisher 2000] Fisher, R. *CV–online: The Evolving, Distributed, Non–Proprietary, On–Line Compendium of Computer Vision. Department of Artificial Intelligence University of Edinburgh, UK.* 2000.

[Flasiński 1991] Flasiński, M. *Syntaktyczne metody rozpoznawania obrazów. Skrypty uczelniane Uniwersytetu Jagiellońskiego, Nr 634.* 1991.

[Flasiński 1992] Flasiński, M. *Strukturalna analiza obrazów za pomocą gramatyk grafowych klasy ETPL(k). Rozprawy habilitacyjne Uniwersytetu Jagiellońskiego, Nr 233.* 1992.

[Fraemling 1996] Fraemling, K. *Modélisation et apprentissage des préférences par réseaux de neurones pour l'aide à la décision multicritère. PhD. thesis : Institut National de Sciences Appliquées de Lyon, Ecole Nationale Supérieur des Mines de Saint–Etienne, France.* 1996.

[Francois & Medioni 1996] Francois, A.R.J., Medioni, G. *Generic Shape Learning and Recognition. In: Proceedings of the International Workshop on Object Representations in Computer Vision, Cambridge, UK.* 1996.

[Friedman 1996] Friedman, J. *Another approach to polychotomous classification. Technical Report, Stanford University.* 1996.

[Fu 1994] Fu, L. Rule Generation from Neural Networks. *IEEE Trans. on SMC*, (24), 1994.

[Fukushima 1975] Fukushima, K. Cognitron: A Self–Organizing Neural Network. *Biological Cybernetics*, 20, 1975, ss. 121–136.

[Fukushima 1980] Fukushima, K. Neocognitron: A Self–Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position. *Biological Cybernetics*, 36(4), 1980, ss. 193–202.

[Geman & Bienenstock 1992] Geman, S., Bienenstock, E. Neural networks and the bias/variance dilemma. *Neural Computation*, 4, 1992.

[Goldberg 1989] Goldberg, D. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison–Wesley, Reading, 1989.

[Goldfarb 1990] Goldfarb, L. On the foundations of intelligent processes – I. An Evolving Model for Pattern Learning. *Pattern Recognition*, 23(6), 1990, ss. 595–616.

[Goldfarb 1992] Goldfarb, L. What is Distance and why do we need the metric model for Pattern Learning ?. *Pattern Recognition*, 25(4), 1992, ss. 431–438.

[Goldfarb, Goldfarb, *et al.* 1995] Goldfarb, L., Goldfarb, L., Bhavsar, V.C., Kamat, V.N. Can a vector space based learning model discover inductive class generalization in a symbolic environment ?. *Pattern Recognition Letters*, 16, 1995, ss. 719–726.

[Gonzalez & Woods 1992] Gonzalez, R.C., Woods, R.E. *Digital Image Processing.* Addison–Wesley, Reading, 1992.

[Grabiński, Wydymus, *et al.* 1989] Grabiński, T., Wydymus, S., Zelias, A. *Metody taksonomii numerycznej w modelowaniu zjawisk społeczno–gospodarczych.* Wydawnictwo Naukowe PWN, Warszawa, 1989.

[Greco, Matarazzo, *et al.* 1998] Greco, S., Matarazzo, B., Słowiński, R., Stefanowski, J. *Induction of decision rules for multicriteria sorting problems. XVI EURO Conference, Bruxelles June 12–15.* 1998.

[Greco, Matarazzo, *et al.* 1999] Greco, S., Matarazzo, B., Słowiński, R. The use of rough sets and fuzzy sets in MCDM. W: Hanne, T., Stewart, T.(red.) *Advances in Multicriteria Decision Making.* Kluwer Academic Publishers, Dordrecht, 1999, ss. 1–59.

[Grossberg 1976] Grossberg, S. Adaptive Pattern Classification and Universal Recording: I. Parallel Development and Coding of Neural Feature Detectors. *Biological Cybernetics*, 23, 1976, ss. 121–134.

[Grossberg 1976] Grossberg, S. Adaptive Pattern Classification and Universal Recording: II. Feedback, Expectation, Olfaction, Illusions. *Biological Cybernetics*, 23, 1976, ss. 187–202.

[Haralick & Shapiro 1985] Haralick, R.M., Shapiro, L.G. Image Segmentation Techniques. *Computer Vision, Graphics and Image Understanding*, 29(1), 1985, ss. 100–132.

[Hayes–Roth, Waterman, *et al.* 1983] Hayes–Roth, F., Waterman, D.A., Lenat, D. *Building expert systems.* Addison–Wesley, Reading, 1983.

[Hertz, Krogh, *et al.* 1993] Hertz, J., Krogh, A., Palmer, R.G. *Wstęp to teorii obliczeń neuronowych.* Wydawnictwa Naukowo–Techniczne, Warszawa, 1993.

[Holland 1975] Holland, J.H. *Adaptation in Natural and Artificial Systems. University of Michigan Press.* 1975.

[Hopfield 1979] Hopfield, J.J. Neural Networks and Physical Systems with Emergent Collective Computational Abilities. W: *Proceedings of the National Academy of Sciences. USA.* 1979, ss. 2554–2558.

[Hough 1962] Hough, P.V.C. *Method and Means for Recognizing Complex Patterns. US Patent 3,069,654, December 18, 1962.* 1962.

[Hunter 1996] Hunter, L. Coevolution learning: Synergistic Evolution of Learning Agents and Problem Representations. W: *Proceedings of Multistrategy Learning Conference.* 1996.

[Jackel & et al. 1995] Jackel, L., et al., Neural–net applications in character recognition and document analysis. W: *Neural–Net Applications in Telecommunications.* Kluwer Academic Publishers, Dordrecht, 1995.

[Jackson 1986] Jackson, P. *Introduction to Expert Systems.* Addison–Wesley, Reading, 1986.

[Jacquet–Lagreze & Siskos 1982] Jacquet–Lagreze, E., Siskos, J. Assessing a set of additive utility functions for multicriteria decision–making, the UTA method. *European Journal of Operational Research*, (10), 1982, ss. 151–164.

[Jain & Karu 1996] Jain, A.K., Karu, K. Learning texture discrimination masks. *IEEE Trans. on PAMI*, 18(2), 1996, ss. 195–205.

[Jaszkiewicz 1999] Jaszkiewicz, A. Improving performance of genetic local search by changing local search space topology. *Foundations of Computing and Decision Sciences*, 24(2), 1999, ss. 77–84.

[Jelonek & Krawiec 1993] Jelonek, J., Krawiec, K. *Zastosowanie sieci neuronowych do wspomagania procesów uczenia sie systemów doradczych. Praca magisterska. Instytut Informatyki, Politechnika Poznańska.* 1993.

[Jelonek, Krawiec, *et al.* 1994a] Jelonek, J., Krawiec, K., Słowiński, R., Stefanowski, J., Szymaś, J. Neural Networks and Rough Sets – Comparison and Combination for Classification of Histological Pictures. W: Ziarko, W.*Rough Sets, Fuzzy Sets and Knowledge Discovery.* Springer–Verlag, 1994, ss. 426–433.

[Jelonek, Krawiec, *et al.* 1994b] Jelonek, J., Krawiec, K., Słowiński, R., Szymaś, J. Rough set reduction of features for picture–based reasoning. W: *Proceedings of The Third International Workshop on RoughSets and Soft Computing.* 1994, ss. 418–425.

[Jelonek, Krawiec, *et al.* 1995] Jelonek, J., Krawiec, K., Słowiński, R., Stefanowski, J., Słowiński, R. Rough Set Reduction of Features for Picture–Based Reasoning. W: Lin, T.Y.*Soft Computing.* Simulation Councils, San Diego, 1995, ss. 89–92.

[Jelonek, Krawiec, *et al.* 1997] Jelonek, J., Krawiec, K., Słowiński, R., Stefanowski, J., Szymaś, J. Computer–assisted diagnosing of neuroepithelial tumours based on clinical and pictorial data. W: Kącki, E.(red.) *Proceedings of The Fourth International Conference 'Computers in Medicine'. Polskie Towarzystwo Informatyki Medycznej.* 1997, ss. 170–175.

[Jelonek & Stefanowski 1997] Jelonek, J., Stefanowski, J. *Using n2–classifier to Solve Multiclass Learning Problems. Research Report RA–011/97, Institute of Computing Science.* 1997.

[Jelonek, Krawiec, *et al.* 1998a] Jelonek, J., Krawiec, K., Słowiński, R., Szymaś, J. Grizzly – An image processing and analysis system oriented towards medical images. *Journal of Decision Systems*, 7(3–4), 1998.

[Jelonek, Krawiec, *et al.* 1998b] Jelonek, J., Krawiec, K., Stefanowski, J. Comparative study of feature subset selection techniques for machine learning tasks. W: *Proceedings of the 7th International Symposium 'Intelligent Information Systems'. Zakopane.* 1998, ss. 68–77.

[Jelonek, Krawiec, *et al.* 1998c] Jelonek, J., Krawiec, K., Słowiński, R. Construction of Textural Features for Classification of Histological Images. W: *Proceedings of the 7th International Symposium 'Intelligent Information Systems'. Malbork.* 1998, ss. 146–149.

[Jelonek, Krawiec, *et al.* 1999] Jelonek, J., Krawiec, K., Słowiński, R., Szymaś, J. Intelligent decision support in pathomorphology. *Polish Journal of Pathology*, 50(2), 1999, ss. 115–118.

[John, Kohavi, *et al.* 1994] John, G.H., Kohavi, R., Pfleger, K. Irrelevant features and the subset selection problem. W: *Machine Learning: Proceedings of the Eleventh International Conference.* Morgan Kaufmann Publishers, Inc., San Francisco, 1994, ss. 121–129.

[Kącki 1997] Kącki, E.(red.) *Proceedings of The Fourth International Conference 'Computers in Medicine'. Polskie Towarzystwo Informatyki Medycznej.* 1997.

[Karim & Alam 1998] Karim, M.A., Alam, M.S. Advances in Recognition Techniques, Part 1. *OptEng*, 37(1), 1998, ss. 7–9.

[Kato, Omachi, *et al.* 1999] Kato, N., Omachi, S., Aso, H., Nemoto, Y. A Handwritten Character Recognition System Using Directional Element Feature and Asymmetric Mahalanobis Distance. *IEEE Trans. on PAMI*, 21(3), 1999, ss. 258–262.

[Kleihues, Buerger, *et al.* 1993] Kleihues, P., Buerger, P.C., Scheithauer, B.W. The new WHO classification of brain tumours. *Brain Pathology*, 3, 1993.

[Kohavi & John 1997] Kohavi, R., John, G.H. Wrappers for feature subset selection. *Artificial Intelligence Journal*, 1–2, 1997, ss. 273–324.

[Kohonen & Sommerfield 1995] Kohonen, T., Sommerfield, D. Feature subset selection using the wrapper method: Overfitting and dynamic search space topology. W: *Proceedings of the First National Conference on Knowledge Discovery and Data Mining (KDD–95).* 1995.

[Kolodner 1993] Kolodner, J. *Case–Based Reasoning.* Morgan Kaufmann Publishers, Inc., San Mateo, 1993.

[Komosiński & Krawiec 2000] Komosiński, M., Krawiec, K. Evolutionary weighting of image features for diagnosing of CNS tumors. *Artificial Intelligence in Medicine*, 19(1), 2000, ss. 25–38.

[Koza 1994] Koza, J.R. *Genetic Programming – 2.* MIT Press, Cambridge, 1994.

[Krawiec & Słowiński 1997] Krawiec, K., Słowiński, R. Learning Discriminating Descriptions from Images. W: *VI International Symposium 'Intelligent Information Systems'. Zakopane.* 1997, ss. 118–127.

[Krawiec, Słowiński, *et al.* 1998] Krawiec, K., Słowiński, R., Szcześniak, I. Pedagodical Method for Extraction of Symbolic Knowledge from Neural Networks. W: Polkowski, L., Skowron, A.(red.) *Rough Sets and Current Trends in Computing. Lecture Notes in Artificial Intelligence (LNAI) vol. 1424.* Springer–Verlag, Berlin, 1998, ss. 436–443.

[Kulikowski 1993] Kulikowski, J.L.(red.) *Selected Topics in Biomedical Image Processing.* IBIB PAN, Warszawa, 1993.

[Laguna, Barnes, *et al.* 1991] Laguna, M., Barnes, J.W., Glover, F.W. Tabu search methods for a single machine scheduling problem. *Journal of Intelligent Manufacturing*, (2), 1991, ss. 63–74.

[Langley, Bradshaw, *et al.* 1983] Langley, P., Bradshaw, G.L., Simon, H.A. Rediscovering chemistry with the BACON system. W: Michalski, R.S., Carbonell, J.G., Mitchell, T.M.*Machine learning: An artificial intelligence approach, volume III.* Morgan Kaufmann Publishers, Inc., San Francisco, 1983.

[Langley 1994] Langley, P. Selection of relevant features in machine learning. W: *Proceedings of the AAAI Fall Symposium on Relevance.* AAAI Press, 1994.

[LeCun & et al. 1989] LeCun, Y., et al., Backpropagation applied to handwritten zip code recognition. *Neural Computation*, (1), 1989, ss. 541–551.

[LeCun & Bengio 1994] LeCun, Y., Bengio, Y. Word–level training of a handwritten word recognizer based on convolutional neural networks. W: *Proc. of the International Conference on Pattern Recognition, volume II.* 1994, ss. 88–92.

[LeCun & Bengio 1995] LeCun, Y., Bengio, Y. Convolutional networks for images, speech, and time–series. W: Arbib, A.(red.) *The Handbook of Brain Theory and Neural Networks.* MIT Press, Cambridge, 1995.

[LeCun & et al. 1995] LeCun, Y., et al., Comparison of learning algorithms for handwritten digit recognition. W: Fogelman, F., Gallinari, P.(red.) *International Conference on Artificial Neural Networks, Paris.* 1995, ss. 53–60.

[Lenat 1977] Lenat, D. On automated scientific theory formation: A case study using the AM program. W: Hayes, J.E., Michie, D., Mikulich, L.I.(red.) *Machine Intelligence 9.* Halsted Press, New York, 1977.

[Liu & Setiono 1997] Liu, H., Setiono, R. A probabilistic approach to feature selection – A filter solution. W: van Someren, M., Widmer, G.(red.) *Proceedings of 9th European Conference on Machine Learning (ECML). Lecture Notes in Computer Science, Vol. 1224.* Springer–Verlag, Berlin, 1997, ss. 319–327.

[Maloof & Michalski 1997] Maloof, M.A., Michalski, R.S. Learning symbolic descriptions of shape for object recognition in X–ray images. *Expert Systems with Applications*, 12(1), 1997, ss. 11–20.

[Mango 1994] Mango, L.J. Computer–assisted cervical cancer screening using neural networks. *Cancer Letters*, 77, 1994, ss. 155–162.

[Marchevsky & Bartels 1994] Marchevsky, A.M., Bartels, P.H. *Image Analysis: A Primer for Pathologists.* Raven Press, New York, 1994.

[Matheus & Rendell 1989] Matheus, C.J., Rendell, L.A. Constructive Induction on Decision Trees. W: *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence.* 1989, ss. 645–650.

[Matheus 1989] Matheus, C.J. A constructive induction framework. W: *Proceedings of the Sixth International Workshop on Machine Learning.* Ithaca, New York, 1989.

[Matheus 1990] Matheus, C.J. Adding Domain Knowledge to SBL through Feature Construction. W: *Proceedings Eighth National Conference on Artificial Intelligence.* AAAI Press / The MIT Press, Cambridge Mass., 1990, ss. 803–808.

[Mehra, Rendell, *et al.* 1989] Mehra, P., Rendell, L.A., Wah, B.W. Principled Constructive Induction. W: *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence.* 1989, ss. 651–657.

[Merz & Freisleben 1997] Merz, P., Freisleben, B. Genetic Local Search for the TSP: New Results. W: *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation.* IEEE Press, 1997, ss. 159–164.

[Michalewicz 1996] Michalewicz, Z. *Genetic Algorithms + Data Structures = Evolution Programs.* Springer–Verlag, Berlin, 1996.

[Michalewicz 1996] Michalewicz, Z. *Algorytmy genetyczne + struktury danych = programy ewolucyjne.* Wydawnictwa Naukowo–Techniczne, Warszawa, 1996.

[Michalski, Carbonell, *et al.* 1983] Michalski, R.S., Carbonell, J.G., Mitchell, T.M. *Machine learning: An artificial intelligence approach, volume III.* Morgan Kaufmann Publishers, Inc., San Francisco, 1983.

[Michalski, Mozetic, *et al.* 1986] Michalski, R.S., Mozetic, I., Hong, J., Lavrac, N. *The multi–purpose incremental learning system AQ15 and its testing application to three medical domains. Proc. of AAAI–86. Philadelphia, USA.* 1986, ss. 1041–1045.

[Michalski & Tecuci 1994] Michalski, R.S., Tecuci, G. *Machine Learning. A Multistrategy Approach. Volume IV.* Morgan Kaufmann Publishers, Inc., San Francisco, 1994.

[Michalski, Rosenfeld, *et al.* 1997] Michalski, R.S., Rosenfeld, A., Duric, Z., Maloof, M.A., Zhang, Q. Learning Patterns in Images. W: Michalski, R.S., Bratko, I., Kubat, M.*Machine Learning and Data Mining. Methods and Applications.* John Wiley & Sons Ltd., 1997, ss. 241–268.

[Michalski, Bratko, *et al.* 1997] Michalski, R.S., Bratko, I., Kubat, M. *Machine Learning and Data Mining. Methods and Applications.* John Wiley & Sons Ltd., 1997.

[Mitchell 1997] Mitchell, T.M. *Machine learning.* McGraw–Hill, 1997.

[Moore & Lee 1994] Moore, A.W., Lee, M.S. Efficient algorithms for minimizing cross validation error. W: Cohen, W.W., Hirsch, H.*Proceedings of the Eleventh International Conference on Machine Learning.* Morgan Kaufmann Publishers, Inc., San Francisco, 1994, ss. 190–198.

[Mulawka 1996] Mulawka, J.J. *Systemy ekspertowe.* Wydawnictwa Naukowo–Techniczne, Warszawa, 1996.

[Narendra & Fukunaga 1997] Narendra, P.M., Fukunaga, K. A branch and bound algorithm for feature subset selection. *IEEE Trans. on Computers*, C(26), 1997, ss. 917–922.

[Ng 1998] Ng, A.Y. On feature selection: Learning with exponentially many irrelevant features as training examples. W: *Proceedings of the Fifteenth International Conference on Machine Learning.* 1998.

[Nieniewski 1998] Nieniewski, M. *Morfologia matematyczna w przetwarzaniu obrazów.* Akademicka Oficyna Wydawnicza, Warszawa, 1998.

[Nishida 1996] Nishida, H. Automatic construction of structural models incorporating discontinuous transformations. *IEEE Trans. on PAMI*, 18(4), 1996, ss. 400–411.

[Ostrowski 1992] Ostrowski, M.(red.) *Informacja obrazowa.* Wydawnictwa Naukowo–Techniczne, Warszawa, 1992.

[Pagallo 1989] Pagallo, G. Learning DNF by Decision Trees. W: *Proc. of the International Joint Conference on Artificial Intelligence.* Morgan Kaufmann Publishers, Inc., San Francisco, 1989.

[Parker 1996] Parker, J.R. *Algorithms for Image Processing and Computer Vision.* Wiley, 1996.

[Pavlidis 1982] Pavlidis, T. *Algorithms for Graphics and Image Processing.* Computer Science Press International, Inc., 1982.

[Pavlidis 1987] Pavlidis, T. *Grafika i przetwarzanie obrazów.* Wydawnictwa Naukowo–Techniczne, Warszawa, 1987.

[Pawlak 1982] Pawlak, Z. Rough Sets. *International Journal of Information & Computer Sciences*, (11), 1982, ss. 341–356.

[Pawlak 1991] Pawlak, Z. *Rough Sets. Theoretical Aspects of Reasoning about Data.* Kluwer Academic Publishers, Dordrecht/Boston/London, 1991.

[Pawlak & Słowiński 1994] Pawlak, Z., Słowiński, R. Rough set approach to multi–attribute decision analysis. Invited Review. *European Journal of Operational Research*, 72, 1994, ss. 443–459.

[Peng & Bhanu 1998] Peng, J., Bhanu, B. Closed–Loop Object Recognition Using Reinforcement Learning. *IEEE Trans. on PAMI*, 20(2), 1998.

[Perrot & Hamey 1991] Perrot, C.G., Hamey, L.G.C. *Object recognition. A survey of the literature. Macquarie Computing Report No. 91–0065C, Macquarie University, Australia.* 1991.

[Pociecha, Podolec, *et al.* 1988] Pociecha, J., Podolec, B., Sokołowski, A., Zając, K. *Metody taksonomiczne w badaniach społeczno–ekonomicznych.* Wydawnictwo Naukowe PWN, Warszawa, 1988.

[Pomerlau 1989] Pomerlau, D. *ALVINN: An autonomous land vehicle in a neural network. Technical Report No. CMU–CS–89–107, Computer Science Dept., Carnegie–Mellon University, Pittsburgh, 1989.*

[Prędki, Słowiński, *et al.* 1998] Prędki, B., Słowiński, R., Stefanowski, J., Susmaga, R., Wilk, Sz. ROSE – software implementation of the rough set theory. W: Polkowski, L., Skowron, A.(red.) *Rough Sets and Current Trends in Computing. Lecture Notes in Artificial Intelligence (LNAI) vol. 1424.* Springer–Verlag, Berlin, 1998, ss. 605–608.

[Quinlan 1979] Quinlan, J.R. Discovering rules by induction from large collections of examples. W: Michie, D.(red.) *Expert Systems in the Micro Electronic Age.* Edinburgh University Press, Edinburgh, 1979.

[Quinlan 1992] Quinlan, J.R. *C4.5: Programs for machine learning.* Morgan Kaufmann Publishers, Inc., San Mateo, 1992.

[Riedmiller & Braun 1992] Riedmiller, M., Braun, H. *RPROP – A Fast Adaptive Learning Algorithm. Technical Report, Univ. Karlsruhe.* 1992.

[Rissanen 1983] Rissanen, J. A Universal Prior for Integers and Estimation by Minimum Description Length. *The Annals of Statistics*, 11(2), 1983, ss. 416–431.

[Ristad & Yianilos 1998] Ristad, E.S., Yianilos, P.N. Learning String–Edit Distance. *IEEE Trans. on PAMI*, 20(5), 1998, ss. 522–532.

[Roy 1990] Roy, B. *Wielokryterialne wspomaganie decyzji.* Wydawnictwa Naukowo–Techniczne, Warszawa, 1990.

[Rumelhart, McClelland, *et al.* 1986] Rumelhart, D.E., McClelland, J.L., the PDP Research Group, *Parallel Distributed Processing. Explorations in the Microstructure of Cognition.* MIT Press, Cambridge, 1986.

[Rutkowski 1996] Rutkowski, L.(red.) *Sieci neuronowe i neurokomputery. Politechnika Częstochowska. Seria Monografie, Nr 40.* 1996.

[Schlimmer 1987] Schlimmer, J.C. Learning and representation change. W: *Proceedings of AAAI–87.* Morgan Kaufmann Publishers, Inc., San Francisco, 1987, ss. 511–515.

[Schyns, Goldstone, *et al.* 1997] Schyns, P.G., Goldstone, R.L., Thibaut, J.–P. The Development of Features in Object Concepts. *Behavioural and Brain Sciences*, , 1997.

[Segen 1994] Segen, J. GEST: A learning computer vision system that recognizes hand gestures. W: Michalski, R.S., Tecuci, G.*Machine Learning. A Multistrategy Approach. Volume IV.* Morgan Kaufmann Publishers, Inc., San Francisco, 1994, ss. 621–634.

[Shortliffe, Perreault, *et al.* 1990] Shortliffe, E.H., Perreault, L.E., Wiederhold, G., Fagan, L.M.(red.) *Medical informatics. Computer Applications in Health Care.* Addison–Wesley, Reading, 1990.

[Słowiński 1992] Słowiński, R.(red.) *Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets Theory.* Kluwer Academic Publishers, Dordrecht, 1992.

[Słowiński 1995] Słowiński, R. Rough set approach to decision analysis. *AI Expert Magazine*, 10(3), 1995, ss. 18–25.

[Słowiński 1997] Słowiński, R. Inteligentne systemy wspomagania decyzji. W: *Materialy 13 Jesiennych Spotkań PTI. Mrągowo.* 1997, ss. 1–22.

[Słowiński 1998] Słowiński, R.(red.) *Fuzzy Sets in Decision Analysis, Operations Research and Statistics. The Handbooks of Fuzzy Sets Series.* Kluwer Academic Publishers, Dordrecht/Boston/London, 1998.

[Stroiński & Węglarz 1997] Stroiński, M., Węglarz, J. Rozwój technik komunikacyjnych. W: *Materiały konferencji Sieci Komputerowe w Nauce, Gospodarce i Administracji POLMAN'97.* 1997, ss. 363–367.

[Szymaś 1997] Szymaś, J. *Wprowadzenie do telepatologii.* Wydawnictwo Poznańskie, Poznań, 1997.

[Tadeusiewicz & Flasiński 1991] Tadeusiewicz, R., Flasiński, M. *Rozpoznawanie obrazów.* Wydawnictwo Naukowe PWN, Warszawa, 1991.

[Tadeusiewicz 1993] Tadeusiewicz, R. *Sieci neuronowe.* Akademicka Oficyna Wydawnicza, Warszawa, 1993.

[Tadeusiewicz 1993] Tadeusiewicz, R. *Problemy biocybernetyki.* PWN, Warszawa, 1993.

[Tadeusiewicz & Korohoda 1997] Tadeusiewicz, R., Korohoda, P. *Komputerowa analiza i przetwarzanie obrazów.* Wydawnictwo Fundacji Postępu Telekomunikacji, Kraków, 1997.

[Thornton 1997] Thornton, C. *Unsupervised constructive learning.* 1997.

[Thrun, Bala, *et al.* 1991] Thrun, S.B., Bala, J.W., Bloedorn, E., Bratko, I., et al., *The MONK's problems: A Performance Comparison of Different Learning Algorithms. Carnegie Mellon University, Pittsburgh, PA.* 1991.

[Tomaszewski 1992] Tomaszewski, T. *Psychologia ogólna.* Wydawnictwo Naukowe PWN, Warszawa, 1992.

[Utgoff 1983] Utgoff, P. Shift of bias for inductive concept learning. W: Michalski, R.S., Carbonell, J.G., Mitchell, T.M.*Machine learning: An artificial intelligence approach, volume III.* Morgan Kaufmann Publishers, Inc., San Francisco, 1983.

[Vafaie & Imam 1994] Vafaie, H., Imam, I.F. Feature selection methods: genetic algorithms vs. greedy–like search. W: *Proceedings of International Conference on Fuzzy and Intelligent Control Systems.* 1994.

[van Laarhoven & Aarts 1987] van Laarhoven, P.J.M., Aarts, E.H.L. *Simulated Annealing: Theory and Practice.* Kluwer Academic Publishers, Dordrecht, 1987.

[Vincke, Gassner, *et al.* 1992] Vincke, P., Gassner, M., Roy, B. *Multicriteria decision–aid.* John Wiley & Sons Ltd., 1992.

[Von Neuman 1966] Von Neuman, J. *Theory of self–reproducing automata. University of Illinois Press.* 1966.

[Wake 1991] Wake, N. Handwritten Alphanumeric Character Recognition by the Neocognitron. *IEEE Trans. on Neural Networks*, 2(3), 1991, ss. 355–365.

[Waterman 1986] Waterman, D.A. *A Guide to Expert Systems.* Addison–Wesley, Reading, 1986.

[Watkins 1989] Watkins, C. *Learning from delayed rewards. Ph.D. thesis, Cambridge University.* 1989.

[Weiss & Kulikowski 1991] Weiss, S., Kulikowski, C.A. *Computer systems that learn: classification and prediction methods from statistics, neural nets, machine learning and expert systems.* Morgan Kaufmann Publishers, Inc., San Francisco, 1991.

[Weiss, Althoff, *et al.* 1994] Weiss, S., Althoff, K.D., Richter, M.M. *Topics in Case–Based Reasoning.* Springer–Verlag, 1994.

[Węglarz 1998] Węglarz, J.(red.) *Recent Advances in Project Scheduling.* Kluwer Academic Publishers, Dordrecht, 1998.

[Whigham 1996] Whigham, P.A. Search Bias, Language Bias and Genetic Programming. W: *Proceedings of Genetic Programming Conference.* 1996.

[Widrow, Rumelhart, *et al.* 1994] Widrow, B., Rumelhart, D.E., Lehr, M.A. The basic ideas in neural networks. *Comm. ACM*, 37(3), 1994, ss. 87–92.

[Wiśniewski & Medin 1994] Wiśniewski, E.J., Medin, D.L. The fiction and nonfiction of features. W: Michalski, R.S., Tecuci, G.*Machine Learning. A Multistrategy Approach. Volume IV.* Morgan Kaufmann Publishers, Inc., San Francisco, 1994, ss. 63–84.

[Wnek & Michalski 1994] Wnek, J., Michalski, R.S. Hypothesis–Driven Constructive Induction in AQ17–HCI: A Method and Experiments. *Machine Learning*, 14, 1994, ss. 139–168.

[Wnek, Kaufman, *et al.* 1995] Wnek, J., Kaufman, K., Bloedorn, E., Michalski, R.S. *Selective induction learning system AQ15c: The method and user's guide. Technical Report MLI 95–4. George Mason University, Machine Learning and Inference Laboratory.* 1995.

[Wolpert & Macready 1995] Wolpert, D., Macready, W.G. *No Free Lunch Theorems for Search. The Santa Fe Institute Technical Report, SFI–TR–95–010.* 1995.

[Wolpert 1996] Wolpert, D. The existence of a priori distinctions between learning algorithms. *Neural Computation*, (8), 1996, ss. 1391–1420.

[Wolpert 1996] Wolpert, D. The lack of a priori distinctions between learning algorithms. *Neural Computation*, (8), 1996, ss. 1341–1390.

[Wong & Chan 1998] Wong, P.K., Chan, Ch. Off–Line Handwritten Chinese Character Recognition as a Compound Bayes Decision Problem. *IEEE Trans. on PAMI*, 20(9), 1998, ss. 1016–1023.

[Yang & Honavar 1998] Yang, J., Honavar, V. Feature subset selection using a genetic algorithm. W: Motoda, H., Liu, H.(red.) *Feature Extraction, Construction, and Subset Selection: A Data Mining Perspective.* Kluwer Academic Publishers, New York, 1998.

[Zabawa 1994] Zabawa, P. *Automatyczne rozpoznawanie liter pisma ręcznego metodą parsingu grafów zaindeksowanych. Politechnika Krakowska. Monografia 172, Seria: InŻynieria elektryczna.* 1994.

[Zeki 1993] Zeki, S. *A Vision of the Brain.* Blackwell Scientific, Cambridge, MA, 1993.

[Zembowicz & Żytkow 1991] Zembowicz, R., Żytkow, J.M. *Automated discovery of empirical equations from data. Proc. ISMIS–91 Symp.* Springer–Verlag, New York, 1991, ss. 429–440.

[Zembowicz & Żytkow 1992] Zembowicz, R., Żytkow, J.M. *Discovery of Equations: Experimental Evaluation of Convergence. Proc. Tenth National Conf. Artif. Intel.* AAAI Press / The MIT Press, Cambridge Mass., 1992, ss. 70–75.

[Zhu 1999] Zhu, S.–Ch. Embedding Gestalt Laws in Markov Random Fields. *IEEE Trans. on PAMI*, 21(11), 1999, ss. 1170–1187.

[Zucker 1976] Zucker, S.W. Region Growing: Childhood and Adolescence. *Computer Vision, Graphics and Image Understanding*, 5, 1976, ss. 382–399.

[Żurada, Barski, *et al.* 1996] Żurada, J., Barski, M., Jędruch, W. *Sztuczne sieci neuronowe.* Wydawnictwo Naukowe PWN, Warszawa, 1996.