Notatki robocze do rozdziału skryptu Uczenie maszynowe i sieci neuronowe, wydawnictwo PP 2004 – spójrz do wydanego skryptu po pełen zweryfikowany tekst

1. Podstawy indukcji drzew

1.1. Wprowadzenie

Metody uczenia się drzew decyzyjnych to najbardziej znane i najczęściej stosowane w praktyce algorytmy indukcji symbolicznej reprezentacji wiedzy z przykładów opisanych za pomocą zbioru atrybutów. Pozwalają one na przybliżenie funkcji klasyfikacyjnych o dyskretnych wartościach wyjściowych odnoszących się do pewnych pojęć, kategorii, klas decyzyjnych zdefiniowanych w oparciu o dostępny zbiór atrybutów. Koncepcja reprezentacji sekwencji warunków wpływających na podejmowaną decyzję z wykorzystaniem struktury drzewa jest stosowana w wielu dziedzinach. Ponadto jest ona w miarę naturalna oraz intuicyjnie zrozumiała dla człowieka. W tym rozdziale przedstawia się najpopularniejsze algorytmy indukcji drzew, takie jak ID3 i C4.5, oraz stosowane w nich miary wyboru atrybutów oparte na mierze entropii informacji. Następnie omawia się zasady budowania drzew binarnych, sposoby traktowania atrybutów liczbowych oraz uwzględniania nieznanych wartości atrybutów. Proponowany przebieg ćwiczenia ma na celu praktyczne zapoznanie się z powyższymi zagadnieniami podczas analizy drzew wygenerowanych z wybranych zbiorów przykładów uczących.

1.2. Reprezentacja drzewa decyzyjnego

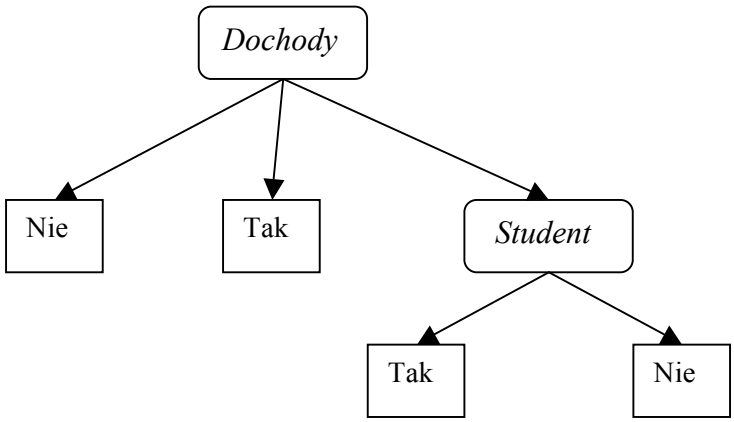
Drzewo decyzyjne składa się z *korzenia*, z którego wychodzą, co najmniej dwie *gałęzie* (krawędzie) do *węzłów* leżących na niższym poziomie. Z każdym węzłem związany jest test (pytanie) sprawdzający wartości atrybutu opisującego przykłady. Dla każdego z możliwych wyników testu (związanego z danym węzłem) prowadzi odpowiadająca mu gałąź do węzła leżącego na niższym poziomie drzewa. Węzły, z których nie wychodzą żadne gałęzie, to *liście*, którym przypisane są klasy decyzyjne (kategorie).

Bardziej formalna definicja (porównaj w [C00,Ha00]) mówi, że drzewo to skierowany graf acykliczny, przy czym krawędzie takiego grafu nazywane są gałęziami, wierzchołki, z których wychodzi, co najmniej, jedna krawędź nazywane są węzłami, a pozostałe wierzchołki – liśćmi. Ponadto przyjmuje się, że w takim grafie istnieje tylko jedna ścieżka między różnymi wierzchołkami.

Drzewo nazywa się *klasyfikacyjnym*, jeśli reprezentuje podział zbioru obiektów (przykładów uczących) na jednorodne klasy [Ga98]. Jego wewnętrzne węzły opisują sposób dokonania tego podziału (w oparciu o wartości atrybutów opisujące obiekty) a liście odpowiadają klasom, do których należą obiekty.

Drzewo może być wykorzystane do klasyfikacji obiektów w następujący sposób: rozpoczynając do korzenia drzewa, sprawdza się wartość atrybutu wymienionego w pytaniu związanym z korzeniem; następnie przesuwamy się do kolejnego wierzchołka wzdłuż gałęzi odpowiadającej wartości atrybutu; proces jest powtarzany dla poddrzewa związanego w tym wierzchołkiem dopóki nie osiągniemy liścia wskazującego klasę. Warto zauważyć, że ze względu na powyższy sposób poruszania się po drzewie podczas klasyfikowania wartości niektórych atrybutów mogą pozostać niewykorzystane.

Rysunek 2.1. ilustruje typowe drzewo decyzyjne wyindukowane za pomocą algorytmu ID3 ze zbioru przykładów uczących reprezentowanych w tabeli 2.1. Pozwala ono na klasyfikacje klientów kupujących komputery w zależności od ich poziomu dochodów i statusu studenckiego. Dla przykładu rozważmy dane o kolejnej osobie (*Dochody* = średnie, *Student* = nie, *Płeć* = kobieta). Sprawdzając testy na wartości atrybutów *Dochody* i nastepnie *Student* dochodzimy przez prawe gałęzie drzewa do liścia (*Kupuje_Komputer* = nie). Zauważmy, że płeć osoby nie ma



Rysunek 2.1 --- Przykładowe drzewo decyzyjne

Zauważmy, że się ścieżki prowadzące od korzenia do liścia drzewa reprezentują koniunkcje pewnych wyrażeń (testów) zdefiniowanych na wartościach atrybutów opisujących przykłady. Jeśli do tej samej klasy prowadzi kilka ścieżek, to tworzą one składniki pewnej alternatywy. Dlatego mówi się, że drzewa decyzyjne pozwa-

lają na nauczenie się pojęć, które można zdefiniować w postaci dysjunkcji takich koniunkcji. Na przykład drzewo przedstawione na rysunku 2.1. pozwala zdefiniować pojęcie klienta pragnącego kupić notebook za pomocą wyrażenia dysjunkcyjnego:

(*Dochody* = średnie) ∧ (*Student* = tak)
∨ (*Dochody* = wysokie)

Drzewo decyzyjne można alternatywnie przedstawić jako zbiór reguł określających przydział obiektów do klas [Mi97]. Każda ścieżka drzewa od korzenia do liścia odpowiada regule. Na przykład drzewo przedstawione na rysunku 2.1. można przekształcić do zbioru czterech reguł:

1. jeśli (*Dochody* = niskie) to (*Kupuje_Komputer* = nie)
2. jeśli (*Dochody* = średnie) ∧ (*Student* = nie) to (*Kupuje_Komputer* = nie)
3. jeśli (*Dochody* = wysokie) to (*Kupuje_Komputer* = tak)
4. jeśli (*Dochody* = średnie) ∧ (*Student* = tak) to (*Kupuje_Komputer* = tak)

Powinno się sprawdzić, czy wszystkie reguły posiadają nie nadmiarowe koniunkcje warunków elementarnych. Można to wykonać przy pomocy operacji chwilowego pominięcia warunku w koniunkcji danej reguły (ang. droping conditions) i porównania tak zmodyfikowanej reguły ze zbiorem przykładów uczących. Jeżeli reguła nadal jednoznacznie klasyfikuje przykładu do właściwej klasy, warunek może być zredukowany, w przeciwnym przypadku należy go odtworzyć w koniunkcji reguły. W powyższym zbiorze reguł można zredukować regułę 2) do następującej postaci, gdyż w tabeli 2.1. wszystkie osoby nie będące studentami równocześnie nie kupiły komputerów:

jeśli (*Student* = nie) to (*Kupuje_Komputer* = nie)

213. Algorytm ID3 indukcji drzew decyzyjnych

Większość algorytmów uczenia się drzew decyzyjnych jest oparta na podobnym heurystycznym schemacie zstępującego konstruowania drzewa (nazwa angielska TDIDT – Top Down Induction of Decision Trees). Jest to rozwiązanie użyte już w pierwszych algorytmach takich jak ID3 i CART [Q86,BF84]. Różnice między konkretnymi algorytmami dotyczą przede wszystkim sposobu wyboru testu dla węzła związanego z oceną jakości podziału zbioru przykładów w węźle, zasad podejmowania decyzji o utworzenia liścia lub węzła, oraz technik uwzględniania różnego rodzaju zaburzeń w opisie przykładów uczących. Podejście to jest dobrze reprezentowane przez najpopularniejszy algorytm o nazwie ID3 [Q79] i jego następcę C4.5 [C45], oba zaproponowane przez J.R.Quinlana.

Podstawowy schemat zstępującego konstruowania drzewa odpowiadający wersji algorytmu ID3 jest przedstawiony poniżej. Zakładamy, że dostępny jest zbiór przykładów uczących *S*. Jeżeli wszystkie przykłady należą do tej samej klasy, to utworzony jest liść i przydzielana jest mu etykieta tej klasy. W przeciwnym przypadku tworzony jest węzeł (w pierwszej iteracji korzeń drzewa) i konieczne jest sformułowanie testu związanego z nim. Rozważa się wszystkie możliwe atrybuty i

ocenia ich przydatność do zbudowania testu prowadzącego do podziału zbioru przykładów *S* na podzbiory jak najbardziej jednorodne w sensie przydziału do klas (w oryginalnej postaci algorytmu ID3 test jest pytaniem o wartość danego atrybutu). Dokonuje się wyboru najlepszego z tych podziałów zgodnie z przyjętą miarą oceny jakości podziału. Rozbudowuje się drzewo poprzez dodanie do węzła gałęzi odpowiadającym poszczególnym wynikom testu. W przypadku algorytmu ID3 gałęzie odpowiadają poszczególnym wartościom $v_1,v_2,..,v_r$ atrybutu *a*. Podzbiór *S* jest podzielony na podzbiory zgodnie z wybranym testem. Następnie używa się rekurencyjnie opisanej procedury dla każdego z tych podzbiorów, budując poddrzewo albo liść, jeśli zajdzie warunek zatrzymania.

funkcja *Buduj_drzewo*
(**argumenty wejściowe**:
S - zbiór przykładów wejściowych,
A – zbiór atrybutów opisujących przykłady {w przypadku ID3 atrybuty są jakościowe lub zdyskretyzowane},
wyjście drzewo decyzyjne);
metoda
Utwórz węzeł *n*; {przy pierwszym wywołaniu korzeń drzewa}
jeśli wszystkie przykłady w *S* należą do tej samej klasy *K* **to**
zwróć *n* jako liść z etykieta klasy *K*;
jeśli zbiór *A* jest pusty **to**
zwróć *n* jako liść z etykietą klasy do której należy większość przykładów w *S*;
w przeciwnym razie
rozpocznij
wybierz atrybut $a \in A$, który najlepiej klasyfikuje przykłady z S zgodnie z przyjętą miarą oceny {dla ID3 *information gain*};
Przypisz węzłowi *n* test wykorzystujący *a*;
dla każdej v_i wartości atrybutu *a* **wykonaj**
dodaj do węzła *n* gałąź odpowiadającą warunkowi ($a = v_i$);
Niech S_i będzie podzbiorem przykładów z *S*, które posiadają wartość v_i dla atrybutu *a*;
jeśli S_i jest pusty **to**
dodaj do gałezi liść z etykietą klasy do której należy większość przykładów w *S*;
w przeciwnym razie
indukuj poddrzewo ***Generuj_drzewo***(S_i , $A - \{a\}$)
koniec;
zwróć drzewo o korzeniu w *n*.

Zasadniczym problemem w algorytmie jest wybór atrybutu do zbudowania testu, w oparciu o który nastąpi w węźle podział zbioru przykładów. Nieformalnie mówiąc, „dobrym" testem jest ten, którego użycie w węźle powoduje skrócenie ścieżki prowadzącej przez ten węzeł do liści wskazujących klasę decyzyjną [C00]. Tak zaś będzie, gdy w każdym podzbiorze związanym z gałęziami wchodzącymi z węzła wszystkie przykłady lub ich większość będzie reprezentowała jedną klasę. Wybór powinien być dokonywany na podstawie miary oceniającej na ile wartości danego atrybutu pozwalają podzielić zbiór przykładów na podzbiory, które charak-

teryzują się maksymalną jednorodnością w zakresie przydziału do klas decyzyjnych. W algorytmie ID3 wykorzystuje się miarę *przyrostu informacji* (ang. information gain). Aby ją zdefiniować zacznijmy od pojęcia *entropii* wywodzącej się z teorii informacji [Mi97].

Niech S będzie zbiorem uczącym zawierającym przykłady należące do jednej z k klas decyzyjnych, oznaczonych przez $K_1,...,K_k$. Niech n będzie liczbą przykładów z S, oraz n_i oznacza liczebność klasy K_i. *Entropia* związana z klasyfikacją zbioru S jest zdefiniowana jako:

$$Ent(S) = -\sum_{i=1}^{k} p_i \cdot \lg_2 p_i \,, \tag{2.1}$$

gdzie p_i jest prawdopodobieństwem, że losowo wybrany przykład z S należy do klasy K_i i jest estymowane się jako n_i/n. Podstawa logarytmu jest równa 2 ponieważ entropia mierzy oczekiwaną liczbę bitów do zakodowania informacji o klasyfikacji losowo wybranego przykładu ze zbioru S (więcej informacji na temat interpretacji teorio-informacyjnej czytelnik może znaleźć w [Ci00,Mi97]). Zauważmy, że dla wykonania obliczeń, w sytuacji gdy którekolwiek $p_i = 0$ przyjmuje się $0 \cdot \log_2 0 = 0$. W przypadku rozważania klasyfikacji binarnej entropia przyjmuje wartości z przedziału [0,1]. Przy czym maksymalna wartość, równa 1, osiągana jest dla $p_1 = p_2 = 0.5$, czyli dla przykładów o równomiernym rozkładzie klas. Najmniejszą wartość, równa 0, przyjmuje entropia, gdy wszystkie przykłady należą do tej samej klasy. Interpretacja wartości entropii jest następująca, im mniejsza wartość entropii, tym w zbiorze S jest większa przewaga przydziału przykładów do jednej z klas nad drugą klasą.

W przypadku użycia atrybutu a do zbudowania testu oblicza się entropie warunkową. Niech atrybut a posiada r różnych wartości $\{v_1,v_2,..,v_r\}$. W algorytmie ID3 test jest konstruowany jako pytanie o wartość atrybutu a, czyli dokonuje się podziału S na podzbiory $\{S_1,S_2,...,S_r\}$, gdzie S_j zawierają przykłady posiadające dla atrybutu a wartość vj (j=1,..,r). Liczebność zbioru S_j jest oznaczona jako n_{Sj}. Entropia podziału zbioru przykładów S ze względu na atrybut a jest zdefiniowana jako:

$$Ent(S|a) = \sum_{j=1}^{r} \frac{n_{S_j}}{n} \cdot Ent(S_j) \,, \tag{2.2}$$

Można stwierdzić, że entropia $Ent(S|a)$ jest średnią ważoną dla entropii poszczególnych podzbiorów S_j. Im mniejsza wartość $Ent(S|a)$, tym większa jednorodność klasyfikacji dla przykładów podzielonych na podzbiory.

Przyrost informacji wynikający z zastosowania atrybutu a do zbudowania testu dzielącego zbiór przykładów uczących S jest zdefiniowany jako różnica:

$$Gain(S,a) = Ent(S) - Ent(S|a), \tag{2.3}$$

$Gain(S,a)$ jest oczekiwaną redukcją entropii spowodowaną znajomością wartości atrybutu a. Innymi słowy, reprezentuje on informację o klasyfikacji przykładów, gdy dana jest wartość atrybutu a opisującego przykłady. Algorytm ID3 obli-

cza wartość przyrostu informacji dla każdego z rozważanych atrybutów. Wybiera się ten atrybut, dla którego zaobserwowano największy przyrost informacji.

Przykład ilustracyjny

W celu ilustracji działania algorytmu ID3, rozważ zbiór przykładów przedstawionych w Tabeli 2.1. Reprezentuje on pewną grupę klientów sklepu z artykułami elektronicznego, z których część decyduje się na zakup notebooka, podczas gdy pozostali nie są gotowi do takiego zakupu. Są oni scharakteryzowani przez zbiór atrybutów *A* zawierający trzy atrybuty wyrażające ocenę poziomu ich dochodów, stwierdzenie faktu czy są studentami oraz ich płeć. Na podstawie wartości tych atrybutów chcemy przewidzieć wartość atrybutu decyzyjnego *Kupuje_Komputer*.

Tabela 2.1. Zbiór przykładów uczących opisujących grupę osób, będących klientami pewnego sklepu elektronicznego. Są oni opisani atrybutami charakteryzującymi ich dochody, status studencki i płeć oraz zaklasyfikowani do dwóch klas na podstawie gotowości do zakupu komputera, będącego notebookiem.

lp.	*Dochody*	*Student*	*Płeć*	*Kupuje komputer*
1	średnie	tak	mężczyzna	Tak
2	średnie	nie	kobieta	Nie
3	wysokie	tak	kobieta	Tak
4	niskie	tak	mężczyzna	Nie
5	niskie	tak	kobieta	Nie
6	średnie	tak	kobieta	Tak
7	niskie	nie	kobieta	Nie
8	średnie	nie	mężczyzna	Nie

W przedstawionym przykładzie występuje binarna klasyfikacja. W związku z tym miara entropii dla zbioru *S* wyraża się następującym wzorem $Ent(S) = -p_{\text{Tak}}\log_2 p_{\text{Tak}} - p_{\text{Nie}}\log_2 p_{\text{Nie}}$. Zbiór ośmiu przykładów składa się z 3 przykładów decyzji *Tak* i 5 przykładów decyzji *Nie*. Odpowiednie prawdopodobieństwa są równe $p_{\text{Tak}} = 3/8$ oraz $p_{\text{Nie}} = 5/8$. Wartość entropii związanej z binarna klasyfikacją rozważanego zbioru przykładów jest następująca $Ent(S) = -(3/8)\cdot\log_2(3/8) - (5/8)\cdot\log_2(5/8) = 0.531 + 0.424 = 0.955$.

Przeanalizujmy sytuacje, gdy atrybut *Dochody* jest użyty do zbudowania korzenia drzewa decyzyjnego. Atrybut ten posiada trzy wartości *niskie*, *średnie*, *wysokie*, które mogą być użyte do podziału zbioru 14 przykładów na trzy podzbiory. Pierwszy podzbiór $S_{\text{niskie}}=\{4,5,7\}$ zawiera trzy przykłady, które wszystkie należą do klasy decyzyjnej *Nie*. Drugi podzbiór $S_{\text{średnie}}=\{1,2,6,8\}$ zawiera po dwa przykłady z obu klas, podczas gdy podzbiór $S_{\text{wysokie}}=\{3\}$ złożony jest z jednego przykładu z klasy *Tak*. Wartość entropii warunkowej ze względu na ten atrybut jest następują-

ca: $Ent(S|Dochody) = (3/8)\cdot Ent(S_{niskie}) + (4/8)\cdot Ent(S_{średnie}) + (1/8)\cdot Ent(S_{wysokie}) = (3/8)\cdot(-0\cdot\log_2 0 - 1\cdot\log_2 1) + (4/8)\cdot(-(1/2)\cdot\log_2(1/2) - (1/2)\cdot\log_2(1/2)) + (1/8)\cdot(-0\cdot\log_2 0 - 1\cdot\log_2 1) = 0 + 0.5 + 0 = 0.5$. Przyrost informacji $Gain(S,Dochody) = Ent(S) - Ent(S|Dochody)$ jest równy $0.955 - 0.5 = 0.455$.

Wartości miar przyrostu informacji wynikających z zastosowania pozostałych atrybutów do budowy korzenia drzew są następujące: $Gain(S,Student) = 0.348$ oraz $Gain(S,Płeć) = 0.004$. Zgodnie z algorytmem atrybut *Dochody* jest najlepszy i zostanie wybrany do stworzenia korzenia drzewa. Podzbiory przykładów przypisane gałęziom odpowiadającym wartościom *niskie* i *wysokie* mają jednoznaczne przydziały do klas decyzyjnych, dlatego te gałęzie można zakończyć liśćmi, etykietowanymi odpowiednio klasami *Nie* i *Tak*. W przypadku podzbioru przykładów $S_{średnie} = \{1,2,6,8\}$ należy rekurencyjnie uruchomić algorytm. Z dwóch rozważanych atrybutów korzystniejszy przyrost informacji pozwala osiągnąć atrybut *Student*, którego wartości jednoznacznie rozdzielają podzbiór przykładów na klasę *Tak* (przykłady 1,6) oraz klasę *Nie* (odpowiednio pozostałe przykłady 2,8). Wynikowe drzewo zamieszczone jest na rysunku 2.1.

1.4. Konstruowanie drzew decyzyjnych dla zróżnicowanych atrybutów

Omawiane dotychczas zagadnienia dotyczyły indukcji drzew decyzyjnych z przykładów opisywanych parami atrybut-wartość, gdzie atrybuty są jakościowe (zdefiniowane na skalach nominalnych) oraz ich wartości są zawsze jednoznacznie zdefiniowane. Ponadto przykłady nie są sprzeczne w sensie przydziału tak samo opisanych przykładów do różnych klas decyzyjnych. W zadaniach odkrywania wiedzy z rzeczywistych danych występują bardziej zróżnicowane lub „zaszumione" opisy przykładów. Dlatego zaproponowano wiele rozszerzeń podstawowych algorytmów indukcji drzew, które wykorzystano w takich algorytmach jak C4.5, Assistant lub CART. Najważniejsze z tych rozszerzeń omawiamy poniżej oraz w kolejnym rozdziale.

1.4.1. Inne miary wyboru atrybutów

Stosowanie miary przyrostu informacji (ang. *gain*) prowadzi do faworyzowania wyboru atrybutów o dziedzinach wielowartościowych w stosunku do wyboru innych atrybutów o dziedzinach złożonych z kilku wartości. Nie jest to pożądana właściwość, zwłaszcza w sytuacjach mocnego zróżnicowania liczności dziedzin atrybutów opisujących analizowane przykłady. Rozważmy skrajny przypadek, gdy pewien atrybut *b*, oznaczający np. datę urodzin, posiada tyle różnych wartości ile jest przykładów uczących. Atrybut ten zostanie wybrany do zbudowania testu w węźle drzewa, gdyż maksymalizuje on wartość miary $Gain(S,b)$. W rezultacie każdy z podzbiorów S_i zawierać bedzie pojedynczy przykład, co doprowadzi do stworzenia płaskiego i równocześnie bardzo szerokiego, drzewa. Takie drzewo odwzorowuje dane uczące, lecz niestety jest mało czytelne dla użytkownika i równocze-

śnie nie jest użyteczne do predykcji klasyfikacji tych przykładów, które nie są reprezentowane w zbiorze uczących. Jeśli rozważymy test z wykorzystaniem atrybutu *b*, który oznaczał pytanie o datę urodzin, to zauważmy, że takie pytanie pozostanie bez odpowiedzi dla nowych przykładów z inną wartością daty niż te, które wystąpiły w zbiorze uczącym.

Jednym ze sposobów uniknięcia tej niedogodności jest zastosowanie innej miary wyboru atrybutów niż miara przyrostu informacji. Quinlan [Q86] zaproponował wykorzystanie dodatkowej miary, która pełni rolę funkcji „kary" dla atrybutów o zbyt licznych dziedzinach. Miara ta, nazywana *podziałem informacji* (z ang. split information), ocenia podział zbioru przykładów ze względu na wartości z dziedziny atrybutu *a*. Jest zdefiniowana w nastepujący sposób:

$$Split(S|a) = \sum_{j=1}^{r} \frac{|S_j|}{|S|} \cdot \log_2 \frac{|S_j|}{|S|}, \tag{2.4}$$

gdzie S_j jest podzbiorem przykładów opisanych *j*-tą wartością atrybutu *a*, *r* jest liczbą różnych wartości w dziedzinie tego atrybutu. Quinlan wykorzystał tą miarę do „normalizacji" przyrostu informacji otrzymując nową miarę oceny jakości testu w węźle nazywaną ilorazem przyrostu informacji (ang. *gain ratio*):

$$Gain\ ratio(S|a) = \frac{Gain(S \mid a)}{Split(S \mid a)} \tag{2.5}$$

Zasada wyboru atrybutu do stworzenia węzła w algorytmie indukcji drzew jest niezmieniona, to znaczy wybiera się atrybut maksymalizujący wartość miary *Gain ratio*. Powyższa miara „wyrównuje" szanse wybranie atrybutów z mniej licznymi dziedzinami w stosunku do atrybutów wielowartościowych. W literaturze można spotkać także inne propozycje alternatywnych miar wyboru atrubutów, porównaj dla przykładu dyskusje przeglądowe w [Mi97, C00, Ga97].

1.4.2. Binaryzacja drzew decyzyjnych

W algorytmie ID3 indukcji drzew przedstawionym w rozdziale 2.3. testy w węzłach drzewa są konstruowane jako pytanie o wartość wybranego atrybutu. Ze względu na to, że rozważa się wyłącznie atrybuty jakościowe, na ogół o niezbyt dużych dziedzinach, liczba krawędzi wychodzących z węzła odpowiada wartościom atrybutu i jest równa przynajmniej dwa. W przypadku analizy bardziej zróżnicowanych danych, powyższe założenia mogą się nie sprawdzać i często modyfikuje się podstawowy schemat algorytmu tak, aby generować *binarne drzewa decyzyjne*.

Binarne drzewo charakteryzuje się tym, że z każdego jego wewnętrznego węzła wychodzą jedynie dwie krawędzie, czyli każdy zbiór przykładów związany z węzłem dzieli się na dwa rozłączne podzbiory. Drzewa binarne stosowane są najczęściej w przypadku klasyfikacji przykładów opisanych *atrybutami ilościowymi*. Ponadto, jak pisze Han [H00], taki rodzaj drzew ogranicza także wystąpienie zjawi-

ska fragmentacji danych, tj. stopniowego podziału zbioru przykładów na coraz mniejsze podzbiory, które mogą zawierać zbyt małą liczbę przykładów.

Konstruowanie binarnych drzew wiąże się z innymi sposobami tworzenia testów do umieszczenia w węźle drzew, tak aby odpowiedzi na test były zawsze dwuwartościowe, np. prawda lub fałsz. Rozważa się dwie możliwe reprezentacje testów do umieszczenia w węźle:

- dla atrybutów jakościowych z r wartościami dokonuje się grupowania wartości należących do podzbioru dziedziny, wtedy test jest reprezentowany w ($a \in G_g$), gdzie G_g jest podzbiorem $\{v_1,v_2,..,v_r\}$; Dla danego atrybutu jest na ogół możliwe skonstruowanie wielu testów w takiej postaci, dlatego stosuje się heurystyczne strategie zachłannego przeszukiwania różnych podzbiorów, aby zredukować koszty obliczeniowe [C45].

- dla atrybutów ilościowych, oraz także porządkowych, testy konstruowane są w postaci ($a \leq \theta$) lub ($a < \theta$) w zależności od przyjętej konwencji. Wartość progu θ jest znajdowana najczęściej poprzez posortowanie różnych wartości atrybutu a i następnie wybór punktów znajdujących się miedzy sąsiednimi wartościami w tym uporządkowaniu jako kandydatów na możliwe progi [C45,KQ02]. Jeżeli dla rozważanego zbioru przykładów S występuję f różnych wartości atrybutu a, to liczba kandydatów do rozważenia jest równa f - 1. W przypadku stosowania do wyboru testu miar opartych na entropii informacji, korzysta się z właściwości podanej w [Fay92]. Według niej jako możliwe wartości progów θ wystarczy rozważać wyłącznie te punkty, które oddzielają przykłady przypisane do różnych klas decyzyjnych. Takie punkty w stosunku do innych punktów, tj. leżących między przykładami należącymi do tej samej klasy, prowadzą do maksymalizacji miary przyrostu informacyjnego. Zainteresowani znajdą w [Mi97] prosty dydaktyczny przykład stosowania tej zasady.

Podziały na dwa podzbiory generowane przed wszystkie rozważane testy dla danego atrybutu są oceniane przy pomocy miar oceny jakości podziału stosowanych w danym algorytmie.

1.4.3. Uwzględnianie niezdefiniowanych wartości atrybutów

Rzeczywiste dane mogą zawierać nieznane (niezdefiniowane) wartości części atrybutów (ang. *unknown values of attributes*) dla niektórych obiektów. Sytuacje takie mogą wynikać z błędów podczas rejestracji danych, zagubienia zapisów lub niedostępności pewnych informacji [Mi97]. Występowanie niezdefiniowanych wartości atrybutów wpływa na sam proces budowy drzewa, jak i użycie go do klasyfikowania nowych lub testowych obiektów.

W literaturze zaproponowano wiele metod radzenia sobie z przykładami, w opisie których występują nieznane wartości atrybutów [Q89,C45]. Część z metod stosowana jest w przetwarzaniu wstępnym danych przed użyciem właściwego algorytmu indukcji. Wiele z tych metod jest ukierunkowane na zastępowanie nieznanej

wartości atrybutu dla określonego przykładu wartością z dziedziny tego atrybutu. Na przykład używa się najczęściej występującej wartości atrybutu, określonej na podstawie przykładów z pełnym opisem lub podzbioru tych przykładów należących do tej samej klasy decyzyjnej co analizowany przykład.

W przypadku systemu C4.5 używa się bardziej skomplikowanej strategii, która jest wykorzystywana jako część wewnętrzna algorytmu. Rozważmy najpierw proces budowy drzewa decyzyjnego. Załóżmy, że S jest zbiorem przykładów, w oparciu o który dokonuje się wyboru atrybutu dla stworzenia testu do węzła drzewa. Niech a będzie potencjalnym atrybutem dla tego testu i S_0 będzie podzbiorem przykładów z S, dla których wartości a są niezdefiniowane. Dla przykładów z S_0 nie jest więc możliwe określenie wyniku zastosowania rozważanego testu. Quinlan zaproponował modyfikacje miar oceny. Obliczane są one na podstawie zdefiniowanej części przykładów (S - S_0) z uwzględnieniem funkcji „kary" zależnej od względnej częstości nieznanych wartości dla tego atrybutu. Miara przyrostu informacji przedstawiona wzorem (2.2) ma w takiej modyfikacji następującą postać:

$$Gain(S,a) = \frac{|S - S_0|}{|S|} Gain(S - S_0,a), \tag{2.6}$$

Jeśli atrybut a został wybrany do stworzenia testu w węźle, to przykład z S_0, opisany nieznanymi wartościami, podlega podziałowi i jest „cząstkowo" przypisany do podzbiorów przykładów S_j odpowiadającym wynikom testu w tworzonym węźle. W takim podziale cząstkowemu przykładowi przypisuje się *wagę* odpowiadającą prawdopodobieństwu wystąpienia określonej wartości atrybutu a. Prawdopodobieństwa te oszacowane są na podstawie częstości występowania różnych wartości atrybutu wśród przykładów w węźle, np. dla podzbioru S_j wagą będzie równa $|S_j| / |S - S_0|$. Tak podzielone przykłady z nieznaną wartością atrybutu a są rozważane w algorytmie do oceny wyboru kolejnych testów w węzłach na niższych poziomach drzewa, przy czym biorą udział w obliczeniach z przydzieloną wagą.

Podobne zasady podziału przykładów (obiektów) stosowane są podczas klasyfikowania nowych obiektów, w opisie, których występują niezdefiniowane wartości atrybutów. W takim przypadku oblicza się rozkład prawdopodobieństwa przydziału obiektu do klas na podstawie sumowania wag podzielonych obiektów klasyfikowanych w różny sposób do węzłów drzewa. Obiekt przydzielany jest do klasy o największym prawdopodobieństwie.

Więcej informacji na powyższe tematy czytelnik może znaleźć w książce [C45].

3. Przeuczenie i redukcja rozmiarów drzew

3.1. Wprowadzenie

Drzewa decyzyjne wygenerowane z przykładów uczących są często wykorzystywane do klasyfikowania nowych obiektów, tzn. określenia ich przydziału do klasy decyzyjnej. W takim zastosowaniu drzewa pełnią rolę klasyfikatorów. Ocenę przydatności drzew do klasyfikowania nowych obiektów dokonuje się najczęściej przez estymację błędu klasyfikowania lub trafności klasyfikowania w odniesieniu do zbioru przykładów testowych – metodologiczne zasady przeprowadzenia takiej oceny omówiono w rozdziale 1. Należy zauważyć, że podczas generowania jak najlepszych drzew - klasyfikatorów można napotkać trudności, zwłaszcza, jeśli rzeczywiste dane zawierają sprzeczne, „zaszumione" lub niekompletne opisy przykładów uczących. W algorytmach indukcji drzew, takich jak C4.5, Assistant lub CART, wprowadzone różne rozszerzenia, które mają na celu polepszyć zdolności klasyfikacyjne drzew w takich sytuacjach [BF84,CKB87,C45]. Niniejszy rozdział omawia najważniejsze z nich, związane ze zjawiskiem przeuczenia i metodami uproszczenia rozmiarów drzew (tzw. przycinania).

3.2. Zjawisko przeuczenia

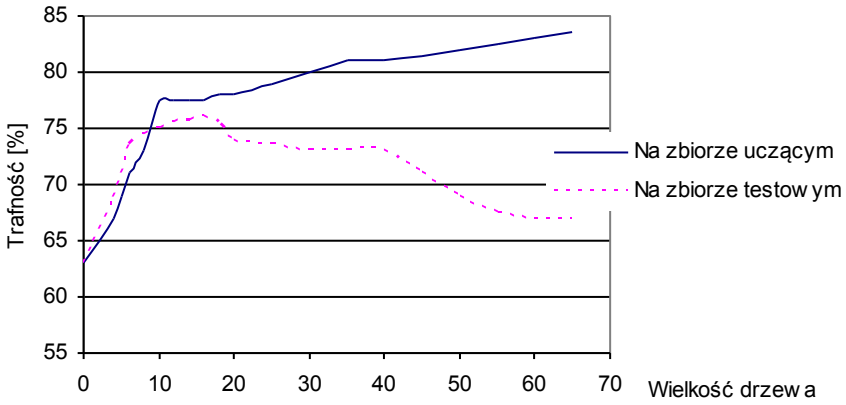
W algorytmach uczenia się drzew decyzyjnych, opisanych w rozdziale 2, poszczególne gałęzie drzewa są rozbudowywane tak głęboko, dopóki przykłady w węźle nie zostaną zaklasyfikowane do pojedynczej klasy decyzyjnej. Powyższy sposób postępowania zakłada niejawnie, że przykłady uczące są zdefiniowane poprawnie. Oznacza to, między innymi, że nie występują **sprzeczne przykłady**, to znaczy przykłady opisane takimi samymi wartościami atrybutów, lecz przydzielone do innych klas. Przy takich założeniach poszukiwanie hipotezy – drzewa spójnego, lub przynajmniej minimalizującego błąd klasyfikowania na zbiorze uczącym może być uzasadnione.

Nie zawsze jednak taki sposób postępowania jest najlepszy. Dotyczy to zwłaszcza analizy danych, które zawierają pewien szum informacyjny. Wówczas zbiór przykładów uczących może zawierać przypadkowe przekłamania wartości atrybutów warunkowych i decyzyjnych [Mi97]. Sprzeczności przykładów mogą też wynikać z niezarejestrowania wszystkich niezbędnych atrybutów potrzebnych do jednoznacznego rozróżnienia obiektów lub nieprecyzyjności przy tworzeniu języka reprezentacji opisu przykładów. Ponadto liczba dostępnych przykładów uczących może być zbyt mała dla stworzenia reprezentatywnej próbki pewnych pojęć decyzyjnych. W takich sytuacjach podczas indukcji drzew decyzyjnych możemy mieć do czynienia z niepożądanym zjawiskiem **przeuczenia** klasyfikatora, nazywanego także **nadmiernym dopasowaniem** (ang. *overfitting*) do zbioru uczącego.

Parafrazując definicję podaną w [Mi97] drzewo decyzyjne (hipoteza) jest nadmiernie dopasowane do danego zbioru uczącego, jeśli istnieje inne drzewo (należą-

ce do przestrzeni dopuszczalnych hipotez) o większym błędzie na tym zbiorze, które mimo tego ma mniejszy błąd rzeczywisty na całym rozkładzie przykładów (tzn. włączając także przykłady nieobecne w zbiorze uczącym)[1].

W przypadku wystąpienia zjawiska nadmiernego dopasowania generowane drzewo odzwierciedla przypadkowe przekłamania w danych lub zbyt szczegółowe regularności nieistotne dla klasyfikacji przykładów. Strategia rozbudowywania gałęzi drzewa w celu jednoznacznego rozróżnienia przykładów z różnych klas prowadzi do bardzo złożonych (o dużych rozmiarach) drzew decyzyjnych. W szczególności, taka strategia zawodzi dla sprzecznych przykładów, gdyż na podstawie ich dostępnych opisów nie jest możliwe zbudowanie gałęzi drzewa kończących się liśćmi z jednoznacznymi przydziałami do klas decyzyjnych. Nadmiernie dopasowane drzewo ma niską zdolność generalizacji przykładów, a sama struktura zbyt dużego i szczegółowego drzewa jest trudna do analizy przez użytkownika. Ponadto, co wydaje się ważniejsze, takie drzewo posiada na ogół niską przydatność klasyfikacyjną dla nowych lub testowych przykładów.



Rysunek 3.1. Przeuczenie podczas indukcji drzew decyzyjnych

Niską przydatność klasyfikacyjną przeuczonego drzewa można zauważyć porównując błąd (lub trafność) klasyfikowania przy użyciu tego samego drzewa równocześnie na odpowiednich zbiorach uczących i testujących (zbiór testujący powinien być niezależny od uczącego). Na rysunku 3.1. przedstawiono ilustrację zjawiska przeuczenia. Na wykresie przedstawiono trafność klasyfikowania na zbiorach uczących i testujących w zależności od wielkości drzewa. Drzewo podlegało stopniowemu uproszczeniu od pełnego drzewa do maksymalnie zredukowanego (do samego korzenia). Zauważmy, że trafność dla zbioru uczącego (linia ciągła) rośnie

[1] Formalną definicję zjawiska przeuczenia oraz dyskusję zgrożeń z nim związanych można znaleźć np. w pozycjach: [C00] – rozdział 2.3.6 poświęcony „brzytwie Ockhmana” lub [Mi97] – rozdział 3.7.1.

monotonicznie wraz z zwiększaniem się wielkości drzewa. Jednakże trafność mierzona na niezależnym zbiorze testowym (linia przerywana) najpierw wzrasta, a później maleje. Pełne drzewo, pomimo, że prowadzi do wysokiej trafności dla zbioru uczącego, jest zdecydowanie niekorzystne za względu na klasyfikacje przykładów testowych. Najlepsza wartość wielkości redukcji rozmiarów drzewa (ze względu na trafność klasyfikowania przykładów testowych) przypada dla drzewa zawierającego około 17 elementów.

Mitchel pisze w [Mi97], że przeuczenie klasyfikatora jest poważnym utrudnieniem zarówno dla indukcji drzew decyzyjnych jak i innych algorytmów uczących się. Opisuje także wyniki studiów eksperymentalnych na „zaszumionych" i niedeterministycznych danych, gdzie zaobserwowano pogorszenie rzeczywistej trafności klasyfikowania o 10-25% w rezultacie przeuczenia drzew generowanych algorytmem ID3.

Zjawiska nadmiernego dopasowania drzewa do danych uczących można unikać, lub przynajmniej minimalizować jego wpływ na zdolność klasyfikowania, poprzez właściwą **redukcję** rozmiarów drzewa, nazywaną także **upraszczaniem** lub **przycinaniem** drzewa (odpowiednik oryginalnego terminu angielskiego "*pruning*"). Idea postępowania polega na tym, że w pełnym drzewie decyzyjnym usuwa się pewne fragmenty (poddrzewa) o niewielkim znaczeniu dla klasyfikacji obiektów. W rezultacie lokalnego usunięcia poddrzewa z pełnego drzewa, z węzłem stanowiącym korzeń tego poddrzewa może być związany zbiór przykładów uczących należących do różnych klas decyzyjnych. Zamieniając taki węzeł na liść decyzyjny, przypisuje się mu etykietę klasy większościowej w danym zbiorze przykładów. Tego typu przekształcenie może spowodować pogorszenie trafności klasyfikowania dla zbioru uczącego, ale może dawać lepsze efekty dla obiektów spoza tego zbioru. W niektórych systemach nie przypisuje się takim liściom etykiety klasy większościowej, lecz traktuje się je jako **liście probabilistyczne**, reprezentujące rozkład prawdopodobieństwa klas.

3.3. Upraszczanie drzew decyzyjnych

Istnieją różne metody upraszczania drzew decyzyjnych (porównaj przeglądy w [Ga98,Mi97]). Mogą one być zaliczone do dwóch podstawowych grup:

- upraszczanie wstępne (ang. *forward pruning* lub *pre-pruning*),
- upraszczanie w pełni zbudowanego drzewa (ang. *post-pruning*).

Metody należące do pierwszej grupy zapobiegają nadmiernemu rozrostowi drzewa zatrzymując tworzenie nowych węzłów, jeśli spełnione zostaną kryteria stopu (zatrzymania) w procedurze zstępującego konstruowania drzewa. Zaproponowano różne miary jakości podziału zbioru przykładów w węźle, dla których przekroczenie pewnej ustalonej wartości granicznej oznacza zakończenie podziału w tym węźle i utworzenie liścia opatrzonego etykietą klasy większościowej.

Przykładowo w algorytmie Assistant [CKB87] stosuje się trzy reguły stopu. Pierwsza z nich wstrzymuje podział zbioru przykładów S (w węźle), gdy większość z nich należy do jednej klasy decyzyjnej. Mierzy się to za pomocą częstości $\alpha = l_{max} / n$, gdzie n jest liczbą przykładów w zbiorze S, l_{max} oznacza liczebność najliczniejszej klasy w S. Reguła stopu może wymagać, np. aby $\alpha \geq 0.9$, co oznacza że proces budowy drzewa będzie zatrzymany w tym węźle, jeśli przynajmniej 90% przykładów w zbiorze S należy do tej samej klasy. Kolejna reguła zatrzymuje podział zbioru S, gdy jego liczebność jest zbyt mała. Do oceny stosowano parametr $\beta = n / n_{all}$, gdzie n_{all} jest liczbą przykładów w całym zbiorze uczącym. Jeśli przyjmie się, np., wartość β równą 0.05, oznacza to, że zbiór przykładów w węźle nie będzie dalej dzielony, gdy jego liczebność spadnie poniżej 5% wszystkich obiektów. Ostatnia z reguł stopu stosowanych w Assistant zatrzymuje rozrost drzewa, jeśli względny przyrost informacji $Gain(S, a)$ wynikający z zastosowania wyboru najlepszego atrybutu był zbyt niski. Jeżeli przynajmniej jedna z powyższych trzech reguł stopu zadziała, zatrzymywano proces podziału w danym węźle.

W przypadku systemu C4.5 stosuje się upraszczanie pełnego drzewa, ale użytkownik ma także możliwość zdefiniowania minimalnej liczebności zbioru przykładów przed podziałem w węźle (domyślna wartość jest równa 2).

W ogólności metody redukcji w trakcie budowy drzewa są prostsze do bezpośredniej interpretacji oraz łatwiejsze w implementacji. Jednak metody upraszczające pełne drzewo są często skuteczniejsze w praktyce. Wynika to z faktu, że w pierwszej grupie metod decyzje podejmowane są na podstawie rozkładu klas decyzyjnych w stosunkowo niewielkim podzbiorze przykładów. Znajomość pełnego drzewa w momencie przycinania, pozwala na bardziej globalną ocenę korzyści z tego wynikających.

Metody należące do drugiej grupy (ang. *post-pruning*) stosują odmienną koncepcję. Najpierw buduje się pełne drzewo spójne ze zbiorem uczącym, które jest potencjalnie bardzo duże, przespecjalizowane i nadmiernie dopasowane do tego zbioru. Następnie stopniowo obcina się pewne fragmenty pełnego drzewa, tj. poddrzewa zakończone liśćmi, tak aby wybrana miara jakości nie ulegała znaczącemu pogorszeniu. Wybór stosowanej miary jest zależny od metody, a sama miara związana jest najczęściej z oszacowaniem błędu klasyfikowania. W niektórych algorytmach miara jest kombinacją błędu klasyfikowania i rozmiaru drzewa.

Typowy schemat przycinania pełnego drzewa (porównaj np. [C00]) nakazuje kolejno przeglądać nieprzycięte węzły w drzewie, zaczynając od najgłębiej położonych (tj. położonych jak najbliżej liści). Węzeł wraz poddrzewem jest tymczasowo redukowany i zastępowany liściem. Następnie oblicza się wybraną miarę oceny (np. oszacowanie rzeczywistego błędu klasyfikowania) dla tak zredukowanego drzewa. Wartość jest porównywana z wartością tej miary oszacowanej dla drzewa, w którym nie dokonano redukcji rozważanego węzła. Jeżeli operacja redukcji pogorszyła wartość miary oceny, to przywraca się tymczasowo usunięte

poddrzewo w tym węźle. W przeciwnym razie, przycięcie drzewa jest zaakceptowane. Operacje są powtarzane dla kolejnych niezredukowanych węzłów.

Niezależnie od stosowanej miary oceny przycinania podstawowe znaczenie ma sposób szacowania jej wartości w zależności od rodzaju zbioru przykładów. Można wyróżnić dwa główne podejścia:

1. Użycie oddzielnego zbioru przykładów, złożonego z przykładów spoza zbioru uczącego, do oceny użyteczności przycięcia węzłów w drzewie.
2. Przycinanie na podstawie wszystkich danych ze zbioru uczącego, ale z wykorzystaniem metod statystycznych do oceny czy redukcja drzewa doprowadzi, lepszego określonym prawdopodobieństwem, do lepszego drzewa.

Pierwsze podejście stosuje się, jeśli dostępna jest dostatecznie duża liczba etykietowanych przykładów, które dzieli się na **zbiór uczący** i **zbiór walidujący** (przycinania) – ang. „*training and validation approach*". Zbiór uczący stosowany jest do generacji drzewa, a walidujący do oceny przydatności redukcji rozmiarów drzewa. Motywacja do takich podejść zakłada, że nawet, jeśli algorytm uczący może podlegać zjawisku przeuczenia w wyniku istnienia losowych błędów czy artefaktów w zbiorze uczącym, to jest mało prawdopodobne, że niezależny zbiór walidujący zawierać będzie takie same losowe fluktuacje. Dlatego stosowanie odpowiednio dużego zbioru walidującego pozwala na bezpieczniejszą realizację procesu przycinania drzewa.

Jeśli nie stosuje się podziału zbioru przykładów, to decyzja o przycięcia może być podejmowana wyłącznie na podstawie heurystycznych oszacowań na zbiorze uczącym. Według jednej z nich wyznacza się pesymistyczne oszacowanie błędu przycinanego węzła, korzystając z modyfikacji rozkładu dwumianowego i porównuje się z błędem liścia, który miałby ten węzeł zastąpić [Q87].

Inne podejście wykorzystuje do podjęcia decyzji o przycięciu węzła miarę oceniającą złożoność kodowania przykładów uczących oraz drzewa decyzyjnego. Podejścia te oparte są na zasadzie minimalizacji długości kodu (ang. *MDL – Minimum Description Length*); Czytelnik może zapoznać się z jej opisem w rozdziale 5.5. książki [C00] lub rozdziale 6 książki [Mi97]. Ocena eksperymentalna skuteczności różnych metod przycinania drzew decyzyjnych przedstawiona jest w pracach [Mg89,MFS95].

Metoda przycinania pełnego drzewa stosowana w systemie C4.5 wykorzystuje wyłącznie zbiór przykładów uczących i jest omówiona szczegółowo w rozdziale czwartym pracy [C45] oraz przedstawiona skrótowo w [KQ02]. W implementacji systemu użytkownik dysponuje możliwością modyfikacji parametru *CF*, który wpływa na stopień uproszczenia drzewa. Mniejsze wartości parametru powodują silniejszą redukcją rozmiarów drzewa, podczas gdy większe zmniejszają rozległość redukcji. Wpływ tego parametru jest silniejszy dla danych o mniejszych rozmiarach. Jak pisze Quinlan w [C45] wartość domyślna, równa 25%, okazała się skuteczna dla wielu problemów klasyfikacyjnych. Zaleca także zmniejszenie jej wartości, gdy użytkownik zauważy, że wartość błędu klasyfikowania zredukowanego

drzewa na zbiorze testowym jest znacznie większa niż oszacowywany rzeczywisty błąd (ang. *estimated error rate*).

Zamiast przycinania pełnego drzewa można także wykonać jego transformacje do zbioru reguł i następnie przeprowadzić uproszczenie tego zbioru. Jak pisze Mitchell w [Mi97], takie podejście jest skuteczną alternatywą dla redukcji drzew w wielu praktycznych problemach. W ogólności upraszczanie zbioru reguł powstałych z drzew obejmuje następujące kroki:

1. Generuj pełne drzewo decyzyjne spójne ze zbiorem uczącym, pozwalając na jego potencjalne nadmierne dopasowanie do zbioru.
2. Wykonaj transformacje struktury drzewa do równoważnego zbioru reguł poprzez utworzenie kolejnej reguły z odpowiedniej ścieżki drzewa. Część warunkowa reguły zawiera koniunkcję wszystkich warunków odpowiadającym testom w węzłach drzewa na ścieżce do korzenia do liścia, a część decyzyjna reguły zawiera klasę decyzyjną wskazywaną przez liść.
3. Uprość (zredukuj) każdą regułę poprzez usunięcie z części warunkowej tych warunków elementarnych, których nieobecność może poprawić oszacowanie zdolności klasyfikacyjnej reguły. Ocena zdolności klasyfikacyjnej reguły dla odróżniania wskazywanej klasy decyzyjnej od innych klas przeprowadzana jest przy pomocy tzw. pesymistycznego oszacowanie trafności reguły – porównaj opis przedstawiony w rozdziale 5 książki [C45].
4. Dla każdej klasy wszystkie uproszczone reguły są przeglądane, aby usunąć te reguły, których obecność nie wpływa na ocenę trafności całego zbioru reguł.
5. Pozostały zbiór reguł jest sortowany w takim sposób, aby uporządkowana lista reguł prowadziła do jak największej trafności klasyfikowanych przykładów. Na ostatnią pozycję w liście wstawia się tzw. **regułę domyślną** z pustą częścią, warunkową wskazującą domyślną klasę decyzyjną.

3.4. Konstruowanie drzew decyzyjnych z wykorzystaniem techniki „okien”

W dotychczasowych opisach różnych sposobów konstruowania drzew zakładano, że algorytm ma dostęp do wszystkich przykładów uczących. Nie oznacza to jednak, że zawsze należy dostarczyć cały zbiór uczący. Rozważa się także możliwość uczenia się drzewa wyłącznie z części tego zbioru i aktualizacji reprezentacji drzewa na podstawie wyników klasyfikowania pozostałych przykładów. Quinlan w [C45] opisuje technikę tzw. **okien** (ang. *windowing*) realizujący taką strategię postępowania. Oryginalną motywacją jej wprowadzenia były ograniczenia pamięci operacyjnej, do której nie było można wprowadzić pełnej reprezentacji zbiorów przykładów o bardzo dużych rozmiarach.

Zasadniczy schemat postępowania jest następujący. Z całego zbioru przykładów losowo wybiera się podzbiór o określonej liczności. Ten podzbiór nazywany jest oknem (ang. *window*) i z niego generowane jest drzewo decyzyjne, które jest na-

stępnie stosowane do klasyfikowania przykładów spoza okna. Następnie, ustala się, które z tych przykładów są błędnie sklasyfikowane za pomocą drzewa. Wybiera się określoną liczbę błędnie sklasyfikowanych przykładów i dodaje do początkowego zbioru uczącego – okna; kolejne drzewo decyzyjne jest generowane z poszerzonego zbioru. Następnie jest ono stosowane do klasyfikowania pozostałych przykładów. Ten sposób postępowania jest powtarzany dopóki nowo wygenerowane drzewo nie będzie dostatecznie dobrze klasyfikować przykładów nienależących do okna. Możliwe jest też zakończenie po określonej liczbie iteracji. W przypadku losowania przykładów do początkowego okna, Quinlan zaleca, aby stosować losowanie prowadzące do równomiernego rozkładu przykładów w klasach decyzyjnych. Jest to szczególnie ważne, jeśli w oryginalnym, pełnym zbiorze rozkład przykładów w klasach decyzyjnych jest niezrównoważony. W przypadku poszerzenia okna o błędnie sklasyfikowane przykłady zaleca asie dodawanie przynajmniej połowy z nich, aby przyspieszyć zbieżność do końcowego drzewa.

W literaturze podaje się także inne uzasadnienia do stosowania tej techniki niż wyłącznie ograniczenia pamięci operacyjnej. Według pierwszej z nich może to przyspieszać konstrukcje końcowego drzewa decyzyjnego, jeśli dane są wolne od szumu informacyjnego i niedoskonałości. W takich sytuacjach można czasami zbudować poprawne drzewo w kilku pierwszych iteracjach na dużo mniejszych zbiorze uczącym niż w przypadku standardowego uruchomienia indukcji na pełnym zbiorze. W książce [C45] podaje się wyniki analizy zbioru *mushroom* klasyfikującego grzyby na jadalne i trujące w zależności od ich cech wyglądu (zbiór zawiera 8124 przykłady). Algorytm uczący C4.5, przy standardowych wartościach parametrów, wygenerował w pierwszej iteracji drzewo, które poprawnie klasyfikowało wszystkie pozostałe przykłady. Z drugiej strony, wskazówki innych autorów, mówią, że technika okien może też wydłużyć proces poszukiwania poprawnego drzewa dla innych rzeczywistych i trudniejszych danych. Inna motywacja dla tej techniki związana jest obserwacją, że w niektórych eksperymentach, przy równomiernym schemacie losowania do początkowego okna, tworzone drzewa były skuteczniejszymi klasyfikatorami niż te otrzymane w standardowy sposób z pełnego zbioru przypadków. Quinlan w [C45] pisze, że zjawiska to mogą być częstsze dla zbiorów danych z niezrównoważonymi klasami decyzyjnymi.

Zauważmy, że możliwe jest także kilkakrotne uruchomienie techniki okien z różnymi oknami losowymi i otrzymanie w efekcie wielu zróżnicowanych drzew decyzyjnych o różnej skuteczności klasyfikacyjnej. Według Quinlana tak otrzymany zbiór zróżnicowanych klasyfikatorów może być eksploatowany przynajmniej na dwa sposoby: wybór jednego najlepszego drzewa lub zastosowanie ich równocześnie do otrzymania pojedynczego klasyfikatora regułowego za pomocą odpowiednio przeprowadzonego procesu upraszczania.

Na koniec należy wspomnieć o innych zaawansowanych metodach indukcji drzew decyzyjnych, rozwijanych dla różnych celów. Jednym z nich jest **przyrostowe konstruowanie drzewa** stosowane w sytuacjach gdzie okresowo pojawiają się nowe przykłady uczące, które należy uwzględnić bez budowania całego drzewa

od nowa (porównaj rozdział 3.5.4 w [C00]). Wypracowano także podejścia do tworzenia wielu zróżnicowanych drzew klasyfikacyjnych i integracji ich w tzw. **złożone klasyfikatory**, które często mają lepszą trafność klasyfikowania niż pojedyncze klasyfikatory. Przykładami takich rozwiązań są techniki o nazwach angielskich bagging, boosting czy pairwise comparison (porównaj dyskusję przedstawioną w [Di00]). Wyzwania eksploracji baz danych o wielkich rozmiarach doprowadziły także do powstania nowych systemów, które nie wymagają wczytania zbioru przykładów do pamięci operacyjnej i charakteryzują się dobrą skalowalnością obliczeń – przegląd propozycji takich jak SPRINT, SLIQ przedstawiono w [Ha00].

Przegląd różnych rozszerzeń algorytmów indukcji drzew dostępny jest w podsumowaniu rozdziału 3 książki [C00], rozdziale 3 w [Mi97], oraz w [KQ02].

Literatura

[BZ92] Bolc L., Zaremba P., Wprowadzenie do uczenia się maszyn, Warszawa, Akademicka Oficyna Wydawnicza, 1992.

[BF84] Breiman L., Friedman J.H., Olshen R.A., Stone C.J., Classification and Regression Trees, Wadsworth Int. Group, 1984.

[CKB87] Cestnik B., Kononenko I., Bratko I., Assistant 86. A knowledge elicitation tool for sophisticated users, w: Bratko I., Lavrac N. (red.), Progress in Machine Learning, Sigma Press, Wilmshow, 1987, s. 31-45.

[C00] Cichosz P., Systemy uczące się, Warszawa, WNT 2000.

[Fay92] Fayyad U.M., Irani K.B., On the handling of continuous-valued attributes in decision tree generation. Machine Learning, 8, 1992, s. 87-102.

[Ga98] Gatnar E., Symboliczne metody klasyfikacji danych, Warszawa, PWN, 1998.

[Ha00] Han J., Kamber M., Data mining: Concepts and techniques, San Francisco, Morgan Kaufmann, 2000.

[KQ02] Kohavi R., Quinlan J.R., Decision Tree Discovery. Rozdział 16.1.3.w: Klösgen W., Żytkow J.M., Handbook of Data Mining and Knowledge Discovery. Oxford Press, 2002, s. 267-276.

[Mi97] Mitchell T., Machine Learning, Boston, Mac-Graw Hill, 1997.

[Q79] Quinlan J.R., Discovering rules by induction from large collection of examples, w: Michie D. (red.), Expert systems in the micro electronic age. Edinburgh Univ. Press, 1979.

[Q86] Quinlan J.R., Induction of decision trees. Machine Learning 1 (1), 1986, s. 81-106.

[Q89] Quinlan J.R., Unknown values in induction, w: Proc. 6th Int. Workshop on Machine Learning, Morgan Kaufmann, San Mateo, CA, 1989, s. 31-37.

[C45] Quinlan J. R., C4.5: Programs for Machine Learning, San Francisco, Morgan Kaufmann, 1993.

[Zyt02] Żytkow J.M., Types and forms of knowledge: Decision Trees. Rodział 5.4. w: Klösgen W., Żytkow J.M., Handbook of Data Mining and Knowledge Discovery. Oxford Press, 2002, s. 54-55.