INSTYTUT BADAŃ SYSTEMOWYCH
POLSKIEJ AKADEMII NAUK

ANALIZA SYSTEMOWA W FINANSACH I ZARZĄDZANIU

Wybrane problemy
Tom 4

Pod redakcją
Jerzego HOŁUBCA

Warszawa 2002

INSTYTUT BADAŃ SYSTEMOWYCH
POLSKIEJ AKADEMII NAUK

ANALIZA SYSTEMOWA W FINANSACH I ZARZĄDZANIU

Wybrane problemy
Tom 4

Pod redakcją
Jerzego HOŁUBCA

Warszawa 2002

Wykaz opiniodawców artykułów zamieszczonych w tomie:

doc. dr hab. Mieczysław KŁOPOTEK

prof. dr hab. Stanisław PIASECKI

prof. dr Elżbieta RAKUS-ANDERSON

prof. dr hab. Andrzej STRASZAK

doc. dr hab. Sławomir WIERZCHOŃ

dr Sławomir ZADROŻNY

Publikacja dofinansowana przez
Agencję Wydawniczo-Poligraficzną "ARGRAF", Warszawa

ISBN 83-85847-74-X

Wydawca: INSTYTUT BADAŃ SYSTEMOWYCH PAN
ul. Newelska 6 01-447 Warszawa

Redakcja: Dział Informacji Naukowej i Wydawnictw
Barbara Kotuszewska, Joanna Runowska, tel. 837-68-22
Druk: Agencja Wydawniczo-Poligraficzna "ARGRAF", Warszawa
Nakład 200 egz., 15 ark.wyd.; 12,8 .ark. druk.

ANALIZA NOWYCH ALGORYTMÓW DYSKRETYZACJI ATRYBUTÓW CIĄGŁYCH

Cezary Kośmider
Zaoczne Studia Doktoranckie IBS PAN

Dyskretyzacja atrybutów ciągłych oferuje szereg zalet dla procesu uczenia maszynowego. Do podstawowych zalet należą: znaczne zmniejszenie czasu uczenia, możliwa poprawa jakości wiedzy w przypadku zaszumionych danych, zwiększenie czytelności generowanej wiedzy. W niniejszej pracy przedstawiamy wprowadzenie do problematyki dyskretyzacji, a następnie doświadczalne potwierdzenie tychże zalet na bazie zaprojektowanych przez autora algorytmów dyskretyzacji z nadzorem. Algorytmy te wykorzystują metodę zstępującą oraz wstępującą, szereg miar stosowanych głównie w indukcji drzew decyzyjnych oraz kilka kryteriów zatrzymania. Część stworzonych algorytmów dyskretyzacji stanowi nowe rozwiązania, a część jest zbliżona do algorytmów spotykanych w literaturze uczenia maszynowego. W pracy przedstawiamy wyniki badań jakości dyskretyzacji, czasu uczenia i rozmiarów hipotez na podstawie wygenerowanych drzew decyzyjnych. W tym celu wykorzystujemy algorytm indukcji drzew decyzyjnych C4.5.

Słowa kluczowe: sztuczna inteligencja, uczenie się maszyn, uczenie się na podstawie przykladów, dyskretyzacja atrybutów ciągłych, dyskretyzacja z nadzorem, miary podziału przestrzeni wartości atrybutów, indukcja drzew decyzyjnych.

1. Wstęp

Dyskretyzacja atrybutów ciągłych jest szczególnym przypadkiem konstruktywnej indukcji, czyli transformacji przestrzeni hipotez systemu uczącego się. Celem jest takie przekształcenie przestrzeni hipotez, aby pojęcie docelowe było w niej reprezentowane oszczędnie i dokładnie oraz aby możliwe było efektywne nauczenie się pojęcia za pomocą danego algorytmu uczenia się. Modyfikacja przestrzeni hipotez ma pozwolić na jej lepsze dopasowanie do charakteru pojęcia docelowego, a także do algorytmu uczącego się. Innymi słowy dyskretyzacja jest zmianą reprezentacji danych, polegającą na zamianie ciągłej przestrzeni atrybutów w przestrzeń dyskretną, zazwyczaj składającą się z niewielu przedziałów. Kluczowe znaczenie ma

struktura zastosowana do reprezentacji zadania uczenia (Amarel 1968). Najlepiej byłoby aby reprezentacja pozwalała algorytmowi uczącemu się jak najdokładniejsze nauczenie się pojęcia w możliwie najkrótszym czasie. Niestety, te dwa cele są ze sobą sprzeczne, gdyż zadanie uczenia stanowi z reguły kompromis pomiędzy jego szybkością a dokładnością.

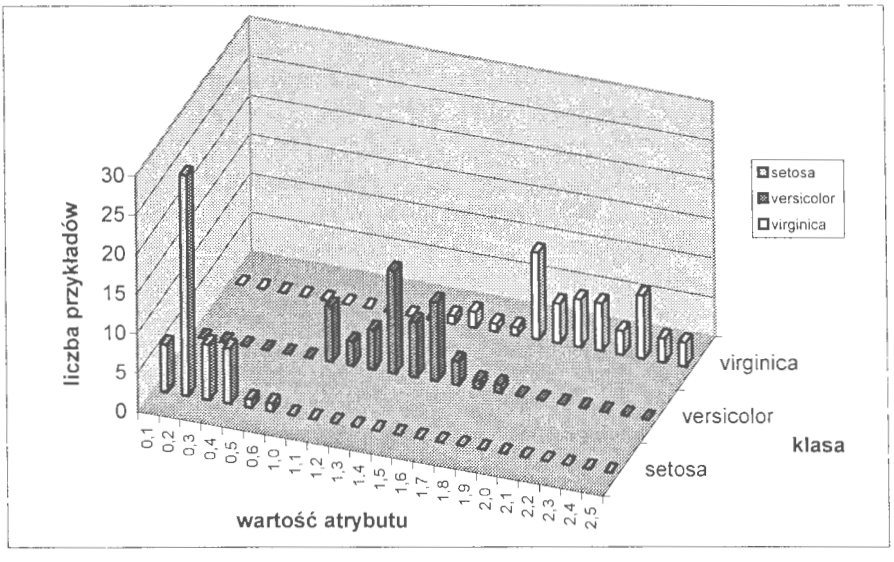
Początkowo proces dyskretyzacji był stosowany z powodu nieumiejętności używania atrybutów ciągłych przez ówczesne algorytmy uczące się. Przykładowo, atrybutów dyskretnych wymagają podstawowe wersje algorytmów indukcji reguł CN2 (Clark i Niblett 1989) i AQ (Michalski 1983). Niektóre późniejsze wersje algorytmów uczących się potrafiły pracować z atrybutami ciągłymi, czego przykładem mogą być ID3 (Quinlan 1986) oraz C4.5 (Quinlan 1993). Co ciekawe dyskretyzacja może być również celowa w przypadku algorytmów, które radzą sobie z atrybutami ciągłymi. Proste algorytmy jak metoda przedziałów równej szerokości (ang. equal width intervals) oraz metoda przedziałów równej częstotliwości (ang. equal frequency intervals) nie wykorzystują informacji o etykietach przykładów. Są one reprezentantami dyskretyzacji *bez nadzoru* (ang. unsupervised, class-blind), wobec czego wytwarzana przez nie dyskretyzacja jest często słabej jakości. Dużo lepszych efektów można spodziewać się po algorytmach wykorzystujących informacje o klasach, które określane są jako metody *z nadzorem* (ang. supervised). Metody te wykorzystują również pewne *miary*, oparte głównie na teorii informacji oraz statystyce (White i Liu 1994), (Kononenko 1995), (Martin 1997), które służą do podziału przestrzeni wartości atrybutów oraz czasami do określenia *kryterium zatrzymania*. Algorytmy te dzielą przestrzeń wartości atrybutu w małą liczbę przedziałów. W tym przypadku użyta miara ma na celu ocenę podziału tej przestrzeni, tak aby w miarę możliwości nie zatracić informacji o klasach.

Oto najpopularniejsze algorytmy dyskretyzacji z nadzorem: D-2 (Catlett 1991), ChiMerge (Kerber 1992), 1R (Holte 1993), Monothetic Contrast Criterion (Merckt 1993) (wersja kryterium łączącego podejście z nadzorem i podejście bez nadzoru), zstępująca minimalizacja entropii (Fayyad i Irani 1993), StatDisc (Richeldi i Rossotto 1995).

Algorytmy klasyfikacji jak np. C4.5 (Quinlan 1993), CART (Breiman i in. 1984) również dokonują dyskretyzacji, lecz nie w kroku *wstępnego przetwarzania* (ang. pre-processing), lecz w trakcie działania algorytmu generacji drzewa. Tak więc w tych algorytmach proces dyskretyzacji jest zintegrowany z procesem indukcji drzew.

Zobaczmy na czym polega proces dyskretyzacji na podstawie przykładu. Oto mały fragment pliku znanego za pewne większości badaczy maszynowego uczenia, zawierającego przykłady dotyczące klasyfikacji odmian irysa. W kolejnych kolumnach mamy odpowiednio wartości atrybutów ciągłych *sepal length, sepal width*, *petal length*, *petal width* oraz w ostatniej kolumnie wartości dyskretnego atrybutu decyzyjnego:

5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
7.0,3.2,4.7,1.4,Iris-versicolor
6.4,3.2,4.5,1.5,Iris-versicolor
6.9,3.1,4.9,1.5,Iris-versicolor
6.3,3.3,6.0,2.5,Iris-virginica
5.8,2.7,5.1,1.9,Iris-virginica
7.1,3.0,5.9,2.1,Iris-virginica



Rysunek 1. Dystrybucja klas dla atrybutu petal width przed procesem dyskretyzacji dla zbioru danych iris

Załóżmy, że chcemy zdyskretyzować atrybut *petal witdth* (4 kolumna) dla całego zbioru *iris*. W tym celu zobaczmy *tablicę kontyngencji* w reprezentacji trójwymiarowej (dystrybucję klas) na poniższym rysunku 1. Widzimy zależności pomiędzy atrybutem ciągłym *petal width*, a dyskretnym atrybutem decyzyjnym (klasą). Na jednej osi mamy wartości atrybutu

ciągłego, które zostały posortowane w porządku rosnącym od 0.1 do 2.5. Na drugiej osi mamy wartości atrybutu decyzyjnego czyli klasy. Na trzeciej została umieszczona liczba przykładów odpowiadająca danej wartości atrybutu ciągłego oraz danej wartości atrybutu decyzyjnego.

Obserwując powyższy rysunek możemy powiedzieć, że dyskretyzacja polega na łączeniu wartości atrybutu ciągłego w przedziały. Proste algorytmy dyskretyzacji czynią to bez obserwacji klas, natomiast zaawansowane algorytmy wykorzystują w tym celu informacje o klasach. W dalszej części pracy prezentujemy efekt dyskretyzacji omawianego atrybutu ciągłego przez prosty algorytm dyskretyzacji bez nadzoru, a następnie efekt dyskretyzacji dokonanej przez algorytm dyskretyzacji z nadzorem.

Zalety dyskretyzacji

- Zmniejszenie liczby wartości atrybutów *przyspiesza* proces uczenia, uczenie staje się bardziej efektywne. Szczególnie znaczne oszczędności czasowe są widoczne w przypadku dyskretyzacji dużych zbiorów danych z wieloma atrybutami ciągłymi. W przypadku atrybutów dyskretnych w procesie uczenia nie jest wymagane *sortowanie* ich wartości, które natomiast jest niezbędne podczas stosowania atrybutów ciągłych.
- Dla danych zaszumionych dyskretyzacja może prowadzić do *poprawy* dokładności generowanych hipotez, gdyż jest pewną formą zapobiegania *nadmiernemu dopasowaniu*.
- Generowane *hipotezy* mogą być bardziej *proste* i tym samym bardziej zrozumiałe dla człowieka.

Rodzaje dyskretyzacji

W literaturze maszynowego uczenia spotykane są następujące podziały metod dyskretyzacji atrybutów ciągłych (Dougherty i in. 1995) na:

- globalne i lokalne,
- z nadzorem i bez nadzoru,
- statyczne i dynamiczne,
- zstępujące i wstępujące.

Metody globalne dyskretyzują jednolicie wartości atrybutów ciągłych dla całej dziedziny, niezależnie od wartości innych atrybutów. Metody *lokalne* dyskretyzują wartości poszczególnych atrybutów w określonych fragmentach dziedziny, które są wyznaczone przez wartości innych atrybutów. Przykładem dyskretyzacji lokalnej jest rekurencyjne budowanie drzewa dla coraz to mniejszych zbiorów przykładów.

Metody *z nadzorem* wykorzystują informację o etykietach przykładów (klasach) w procesie dyskretyzacji. Metody *bez nadzoru* nie wykorzystują informacji o etykietach przykładów podczas dyskretyzacji.

Metody *statyczne* przeprowadzają jeden cykl dyskretyzacji dla każdego atrybutu i określają maksymalną liczbę przedziałów dla każdego atrybutu niezależnie od innych atrybutów. Metody *dynamiczne* przeprowadzają bezpośrednie przeszukiwanie przestrzeni możliwych liczb przedziałów dla wszystkich atrybutów jednocześnie, tym samym biorąc pod uwagę współzależności pomiędzy atrybutami.

Ponieważ metody *zstępująca* i *wstępująca* są podstawą stworzonych algorytmów dyskretyzacji, wobec tego ich dokładne omówienie zamieszczono w dalszej części pracy.

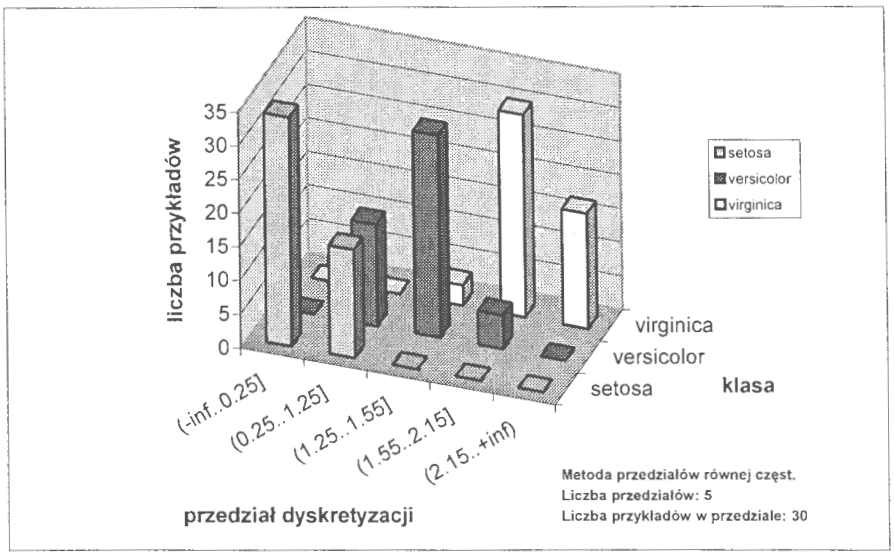
Proste algorytmy dyskretyzacji bez nadzoru

Z prostych algorytmów dyskretyzacji bez nadzoru najbardziej znane są dwa następujące podejścia:

- Metoda przedziałów równej szerokości (ang. equal width intervals) polega na podzieleniu dziedziny wartości atrybutu pomiędzy minimalne i maksymalne wartości w N przedziałów o równych szerokościach (N jest parametrem regulowanym przez użytkownika). Zatem, jeżeli A i B są odpowiednio dolną i górną wartością atrybutu, wtedy przedziały będą miały szerokość W=(B-A)/N i granice przedziału będą co A+W, A+2W,...,A+(N-1)W.
- Metoda przedziałów równej częstotliwości (ang. equal frequency intervals) – granice przedziału są wybierane tak, że każdy przedział zawiera w przybliżeniu tę samą liczbę przykładów trenujących; tak więc, jeżeli N=10, każdy przedział powinien zawierać w przybliżeniu 10% przykładów.

Oto przykład dyskretyzacji dokonanej przez metodę przedziałów równej częstotliwości dla zaprezentowanego wcześniej atrybutu *petal width* (rysunek 2):

Jak możemy zauważyć algorytm przedziałów równej częstotliwości nie analizował klas przykładów podczas dyskretyzacji. Przykładem tego jest przedział (0.25-1.25], gdzie można było uniknąć zgrupowania klas *setosa* oraz *versicolor*. Jednakże algorytm ten dokonał takiego zgrupowania gdyż jedynym warunkiem, który algorytm miał spełnić była w przybliżeniu taka sama liczba przykładów w każdym z przedziałów (w omawianym przykładzie 30).



Rysunek 2. Dystrybucja klas dla atrybutu petal width po procesie dyskretyzacji dla zbioru danych iris

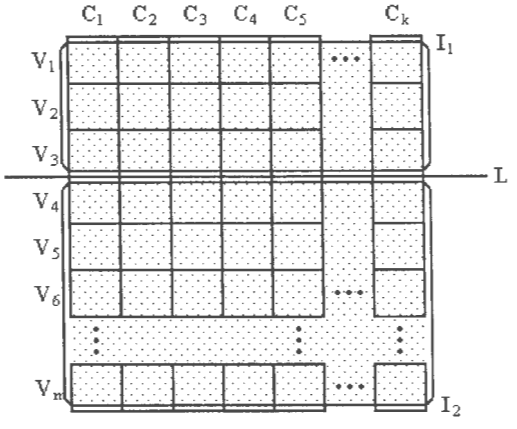
2. Opracowane algorytmy dyskretyzacji

Zaprojektowane przez autora algorytmy dyskretyzacji to algorytmy dyskretyzacji *z nadzorem*. Ze względu na ich architekturę niezbędne jest zwrócenie uwagi na omówione poniżej kwestie dotyczące metod dyskretyzacji (*zstępującej* i *wstępującej*), *miar* oraz *kryteriów zatrzymania*.

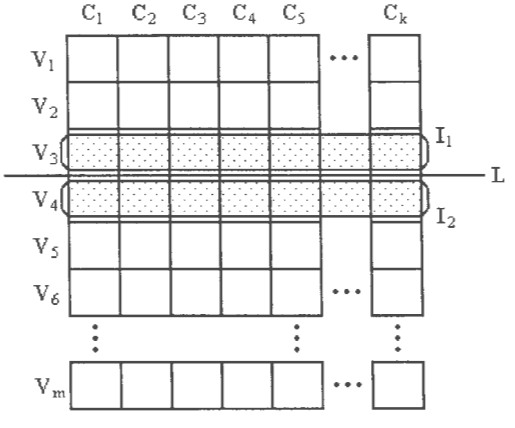
Rysunek 3 prezentuje efekt binarnego podziału wartości atrybutu dla metody zstępującej, natomiast rysunek 4 sytuację przed połączeniem dwóch przedziałów dla metody wstępującej. W obu przypadkach widzimy tablicę kontyngencji, której wiersze odpowiadają wartościom atrybutu mającego ulec dyskretyzacji oznaczonym literą V, natomiast kolumny odpowiadają wartościom atrybutu decyzyjnego oznaczonym literą C. Mamy m wartości atrybutu i k klas.

W przypadku algorytmu *zstępującego* dla każdego atrybutu następuje posortowanie jego wartości (rysunek 3). Początkowo cały zakres posortowanych wartości jest traktowany jako jeden duży przedział. Następnie badane są wszystkie możliwe punkty podziału, spośród których wybierany jest jeden, który osiąga progową (w aktualnej implementacji – *maksymalną*) wartość miary oceniającej podział przestrzeni wartości atrybutu. Załóżmy, że ze wszystkich możliwych punktów podziału został wybrany punkt oznaczony literą L (rysunek 3). Wybrany punkt podziału

dzieli całą przestrzeń wartości atrybutu na dwa zakropkowane podprzedziały oznaczone symbolami I_1 oraz I_2. Cykl jest powtarzany dla kolejnych podprzedziałów, aż do spełnienia określonego kryterium zatrzymania. Proces dyskretyzacji jest powtarzany dla wszystkich atrybutów występujących w zbiorze trenującym.



Rysunek 3. Metoda zstępująca – sytuacja po binarnym podziale wartości atrybutu



Rysunek 4. Metoda wstępująca – sytuacja przed połączeniem dwóch przedziałów

W przypadku algorytmu *wstępującego* również następuje posortowanie wartości dyskretyzowanego atrybutu. Początkowo w każdym przedziale znajduje się jeden przykład (oczywiście jeżeli nie było takich samych wartości atrybutu). Następnie badane są wszystkie możliwe punkty

łączenia dla dwóch sąsiadujących przedziałów. Spośród punktów łączenia wybierany jest jeden, który osiąga progową (w aktualnej implementacji – *minimalną*) wartość miary oceniającej podział przestrzeni wartości atrybutu. Załóżmy, że ze wszystkich możliwych punktów łączenia został wybrany jeden, oznaczony literą L na rysunku 4. Wobec tego nastąpi połączenie dwóch zakropkowanych przedziałów oznaczonych symbolami I_1 oraz I_2. Cykl powtarza się aż do spełnienia określonego kryterium zatrzymania. Proces dyskretyzacji jest powtarzany dla wszystkich atrybutów występujących w zbiorze trenującym.

Miary są sercem działania wielu algorytmów maszynowego uczenia. W algorytmach dyskretyzacji z nadzorem heurystyki te dokonują oceny możliwych podziałów przestrzeni wartości atrybutów przy wykorzystaniu informacji o etykietach przykładów. Ich jakość ma zasadniczy wpływ na jakość dyskretyzacji generowanej przez algorytmy dyskretyzacji z nadzorem. Mogą być one zastosowane również w wielu innych problemach uczenia maszynowego, np.: w algorytmach indukcji drzew decyzyjnych, algorytmach indukcji reguł, do określania kryteriów zatrzymania. Ponieważ w przyjętej implementacji dokonywany podział wartości atrybutów jest podziałem binarnym, wobec tego problem *obciążenia* (ang. bias) miar dotyczący preferencji atrybutów wielowartościowych wydaje się być nieistotny (Mantaras 1989, 1991), (Liu i White 1994), (White i Liu 1994), (Kononenko 1995), (Martin 1997). Jednakże, miary minimalizujące wspomniane obciążenie przyrostu informacji, mianowicie *współczynnik przyrostu* oraz *miara odległości D*, powodują że dokonana za ich pomocą dyskretyzacja dla atrybutów binarnych różni się od dyskretyzacji dokonanej przy wykorzystaniu *przyrostu informacji*. Wobec tego współczynnik przyrostu oraz miara odległości D zostały również zaimplementowane i poddane badaniom.

Podobnie jak użyta miara powinna dokonywać właściwego podziału przestrzeni wartości atrybutów, tak też *kryterium zatrzymania* powinno we właściwym momencie zatrzymać pracę algorytmu, aby pogodzić kompromis pomiędzy dokładnością, a szybkością procesu uczenia. Zbyt duża liczba utworzonych przedziałów podaje w wątpliwość sens wykorzystania algorytmu dyskretyzacji, gdyż czas uczenia niewiele zmaleje w stosunku do *surowych danych* (ang. raw data), wygenerowana hipoteza (np. drzewo decyzyjne) będzie zbyt duża i będzie się cechować niską jakością. Natomiast zbyt mała liczba utworzonych przedziałów spowoduje, że czas uczenia znacznie się zmniejszy, lecz przy dużej stracie dokładności.

Na bazie omówionych pojęć i zależności możemy przedstawić poglądowy schemat zaprojektowanych algorytmów dyskretyzacji (multidyskretyzatora) przedstawiony na rysunku 5:

Rysunek 5. Schemat stworzonego multidyskretyzatora

Zaprojektowany *multidyskretyzator* został oparty o:

Metody:

- *zstępującą* (ang. top-down),
- *wstępującą* (ang. bottom-up).

Miary:

- *przyrost informacji* (ang. information gain) (Quinlan 1986),
- *współczynnik przyrostu* (ang. gain ratio) (Quinlan 1986),
- *miara odległości D* (ang. distance measure D) (Mantaras 1989, 1991),
- *chi kwadrat* (ang. chi square) (Agresti 1990),
- *statystyka G* (ang. G statistic) (Mingers 1987, 1989),
- *indeks gini* (ang. gini index) (Breiman i in. 1984),
- *związek* (ang. relevance) (Baim 1988),
- *średnia waga ewidencji* (ang. average absolute weight of evidence) (Michie 1989),
- *miara J* (ang. J measure) (Smyth i Goodman 1990),
- *rozdwajanie* (ang. twoing) (Breiman i in. 1984),
- *zasada minimalnej długości kodu* (ang. MDL - minimum description length) (Rissanen 1983), (Li i Vitanyi 1993), (Kononenko 1995),
- *miara ortogonalności* (ang. orthogonality metric, angular disparity) (Fayyad i Irani 1992).

Kryteria zatrzymania:

- *próg wartości miary*,
- *próg liczby przedziałów*,
- *próg liczby przykładów w przedziale*,
- *zerowa wartość miary dla wszystkich punktów podziału.*

Wszystkie przedstawione powyżej parametry mogą być dowolnie ze sobą zestawiane, przez co istnieje możliwość generacji szerokiego wachlarza algorytmów dyskretyzacji.

Algorytmy możliwe do wygenerowania przez multidyskretyzator mogą być zaklasyfikowane do następujących grup na podstawie podziału algorytmów dyskretyzacji przedstawionego we wstępie:

- dyskretyzacja globalna,
- dyskretyzacja z nadzorem,
- dyskretyzacja statyczna,
- dyskretyzacja zstępująca i wstępująca.

Dodatkowa dosyć istotna informacja to fakt *binarnego* podziału przestrzeni wartości atrybutów dla metody zstępującej / łączenia *dwóch* przedziałów w każdej iteracji dla metody wstępującej.

Elastyczna konfiguracja multidyskretyzatora pozwala na generację i analizę nowych algorytmów dyskretyzacji oraz wariantów zbliżonych do algorytmów dyskretyzacji spotykanych w literaturze maszynowego uczenia.

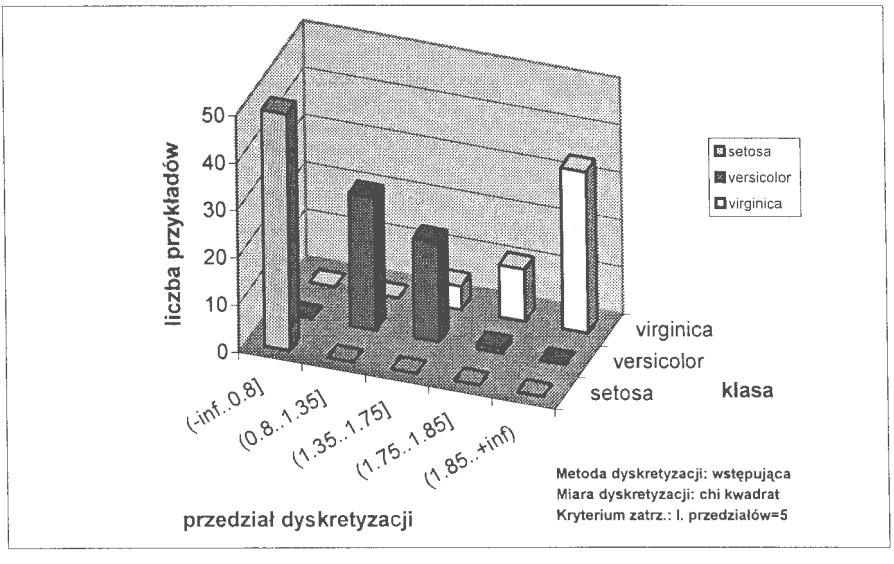
Podobieństwa do istniejących algorytmów dyskretyzacji

Algorytm zstępującej minimalizacji entropii (Fayyad i Irani 1993) wykorzystuje metodę *zstępującą* oraz miarę *przyrostu informacji* w procesie dyskretyzacji, co stanowi główne podobieństwo w stosunku do multidyskretyzatora. Jednakże zasadnicza różnica w rozwiązaniu Fayyad'a i Irani'ego polega na dyskretyzacji zbioru danych podczas generacji drzewa. Tak więc mamy tutaj do czynienia z dyskretyzacją *lokalną*, gdzie podział wartości atrybutów jest generowany dla coraz to mniejszych podzbiorów danych w przeciwieństwie do dyskretyzacji globalnej stosowanej przez multidyskretyzator. Następna różnica to użycie w zstępującej minimalizacji entropii *kryterium zatrzymania* wykorzystującego *MDL*. MDL jest także wykorzystywany przez multidyskretyzator, jednakże do podziału wartości atrybutów. Po niewielkich modyfikacjach multidyskretyzatora możliwe byłoby również wykorzystywanie MDL jako kryterium zatrzymania.

Na podstawie informacji ukazanych w artykule Catlett'a przedstawiającego jego algorytm dyskretyzacji D-2 (Catlett 1991) możemy powiedzieć, że multidyskretyzator może w dużym stopniu zbliżyć się do algorytmu D-2. D-2 używa metody *zstępującej* i miary *przyrostu informacji*. *Kryterium zatrzymania* zostało zdefiniowane przez Catlett'a jako minimalnie 14 przykładów w przedziale oraz maksymalnie 8 przedziałów dyskretyzacji.

Algorytm ChiMerge wykorzystuje metodę *wstępującą* oraz miarę *chi kwadrat*. *Kryterium zatrzymania* stanowi próg wartości miary oraz próg liczby przedziałów (Kerber 1992). ChiMerge może być w pełni zasymulowany przez multidyskretyzator.

Oto ostatni z serii wykres prezentujący dyskretyzację atrybutu *petal width* ze zbioru *iris*, tym razem dokonaną przez jeden z wielu stworzonych algorytmów, tj. algorytm oparty o metodę *wstępującą* i miarę *chi kwadrat*:



Rysunek 6. Dystrybucja klas dla atrybutu petal width po procesie dyskretyzacji dla zbioru danych iris

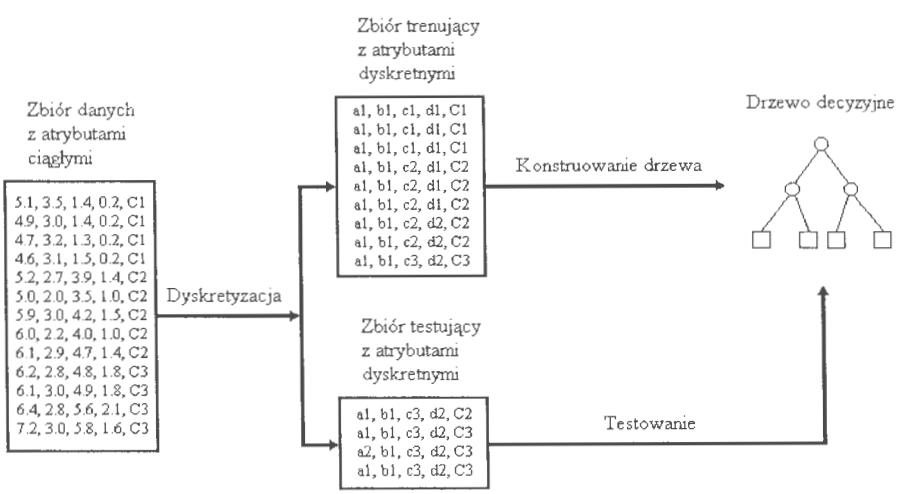
W tym wypadku (rysunek 6) widać „inteligencję" algorytmu. Dzięki użyciu miary (w tym przypadku chi kwadrat) efekt dyskretyzacji jest bardzo dobry. Klasy dla każdego przedziału są wyraźnie od siebie odseparowane. Warto zauważyć, że wynik dyskretyzacji zaprezentowany na powyższym rysunku jest zbliżony dyskretyzacji dokonanej przez ChiMerge. Wynik byłby identyczny po dodaniu kryterium zatrzymania stanowiącego próg wartości miary.

Pewne podobieństwa można również zauważyć w stosunku do algorytmów indukcji drzew C4.5 (Quinlan 1993), CART (Breiman i in. 1984). W tym przypadku podobieństw jest mniej i polegają one w głównej mierze na wykorzystaniu tych samych miar, tj. *współczynnika przyrostu* (C4.5), *rozdwajania*, *indeksu gini* (CART).

3. Idea badań opracowanych algorytmów

Należy zdawać sobie sprawę z tego, że opracowane algorytmy dyskretyzacji atrybutów ciągłych mogą być ocenione pośrednio poprzez efekt pracy algorytmów uczących się. Wobec tego ocena algorytmów dyskretyzacji jest zależna od wielu czynników, np. od:

- sposobu utworzenia zdyskretyzowanych zbiorów trenującego i testującego,
- sposobu podziału zbioru danych na zbiór trenujący i testujący np. dziesięciokrotna walidacja (ang. 10-fold cross-validation),
- sposobu działania wybranego algorytmu indukcji drzew decyzyjnych,
- wyboru parametru oceniającego jakość klasyfikatora (np. współczynnik błędu).



Rysunek 7. Ogólny schemat badań

Powyższy rysunek 7 przedstawia ogólny schemat badań. Zbiór danych z atrybutami ciągłymi był podawany na wejście każdego ze stworzonych algorytmów dyskretyzacji atrybutów ciągłych. Po procesie dyskretyzacji zbiór danych był dzielony na zbiór trenujący oraz zbiór testujący. *Zbiór*

trenujący służył do utworzenia drzewa decyzyjnego, natomiast *zbiór testujący* był wykorzystany do testowania tegoż drzewa.

Do badań wykorzystano algorytm C4.5 (wersja 8) (Quinlan 1993) z pakietu uczenia maszynowego WEKA 3.1.8 (www.cs.waikato.ac.nz). Algorytm ten jest od wielu lat z powodzeniem wykorzystywany przez badaczy maszynowego uczenia. Stał się już pewnym punktem odniesienia, wobec którego badacze przedstawiają wyniki swoich prac. C4.5 został bardzo dobrze przetestowany, na jego bazie powstało dużo odmian zaspokajających szeroki wachlarz wymagań.

Obserwując powyższy rysunek 7 badania jakości wiedzy wygenerowanej przez algorytmy drzew decyzyjnych przeprowadzono poprzez ocenę jakości *drzewa decyzyjnego* (testowanie). Parametrem oceniającym był *współczynnik błędu* (ang. error ratio) określony następująco:

$$ER = \frac{I+U}{N} \cdot 100\%$$

gdzie:

I – liczba niepoprawnie zaklasyfikowanych przykładów,
U – liczba nie zaklasyfikowanych przykładów,
N – liczba wszystkich przykładów.

Podczas oceny jakości wiedzy zastosowano warstwowaną dziesięciokrotną walidację (ang. stratified 10-fold cross-validation). *Warstwowana dziesięciokrotna walidacja* polega na podziale zbioru danych na 10 równych części o podobnej dystrybucji klas. Każda część jest po kolei używana jako zbiór testujący dla klasyfikatora wygenerowanego z pozostałych 9 części (zbiór trenujący) (Quinlan 1996). Na powyższym rysunku 7 ze względu na czytelność nie odnotowano zastosowania dziesięciokrotnej walidacji.

Badania rozmiarów drzew przeprowadzono poprzez sumowanie węzłów oraz liści wygenerowanego *drzewa decyzyjnego*. Na wspomnianym rysunku 7 *węzły* zostały oznaczone kołami, natomiast *liście* zostały oznaczone kwadratami.

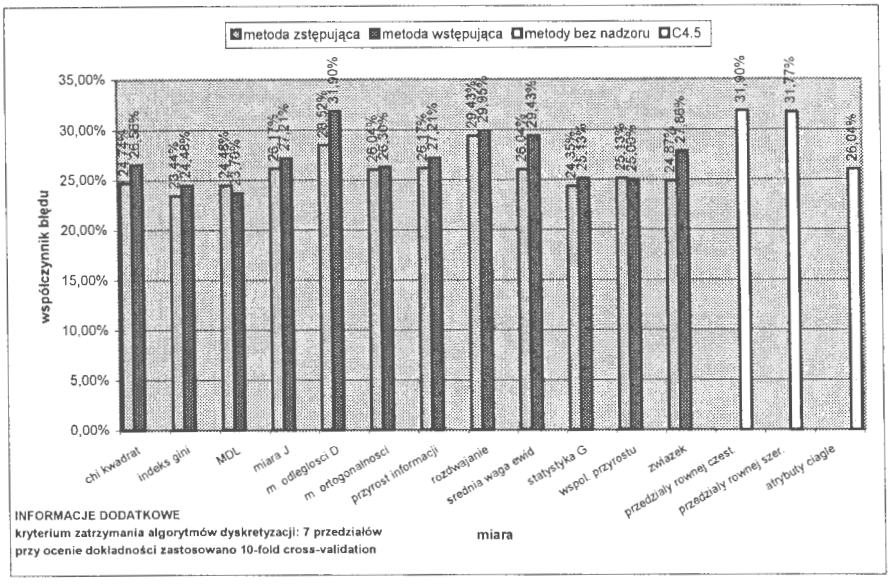
Badania czasu uczenia drzew decyzyjnych polegały na pomiarze czasu budowy *drzewa decyzyjnego*. Na rysunku 7 ten proces został określony jako konstruowanie drzewa. Czas uczenia drzew został wyrażony w milisekundach.

W badaniach wykorzystano zbiory danych *iris*, *glass*, *diabetes*, *vehicle* wzięte ze składnicy uczenia maszynowego UCI.

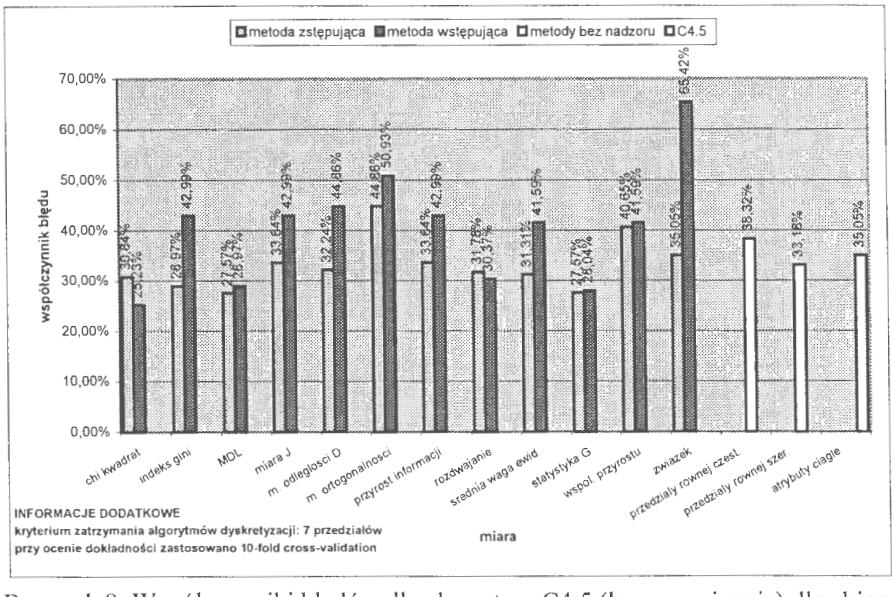
Badania jakości drzew decyzyjnych

W pierwszej części badań jakości drzew decyzyjnych (rysunki 8, 9, 10, 11; zbiory *diabetes*, *glass*, *iris*, *vehicle*) przyjęto, że każdy z atrybutów musi przyjąć stałą liczbę przedziałów dyskretyzacji (w tym przypadku 7) ze względu na potrzebę porównywalności metody *zstępującej* i *wstępującej* oraz *miar* oceniających podział przestrzeni wartości atrybutów. Głównym celem tej serii badań było rzetelne porównanie opracowanych algorytmów dyskretyzacji, a nie stworzenie najlepszego dyskretyzatora. Do badań wykorzystano algorytm indukcji drzew decyzyjnych C4.5 z wyłączonym przycinaniem. Algorytm *przycinania* został wyłączony, gdyż w pewnym stopniu mógłby zaburzać porównanie metod oraz miar. W celach porównawczych zamieszczono również wyniki dyskretyzacji algorytmu C4.5 dla atrybutów ciągłych oraz wyniki metod dyskretyzacji bez nadzoru tj. metody przedziałów równej szerokości oraz metody przedziałów równej częstotliwości.

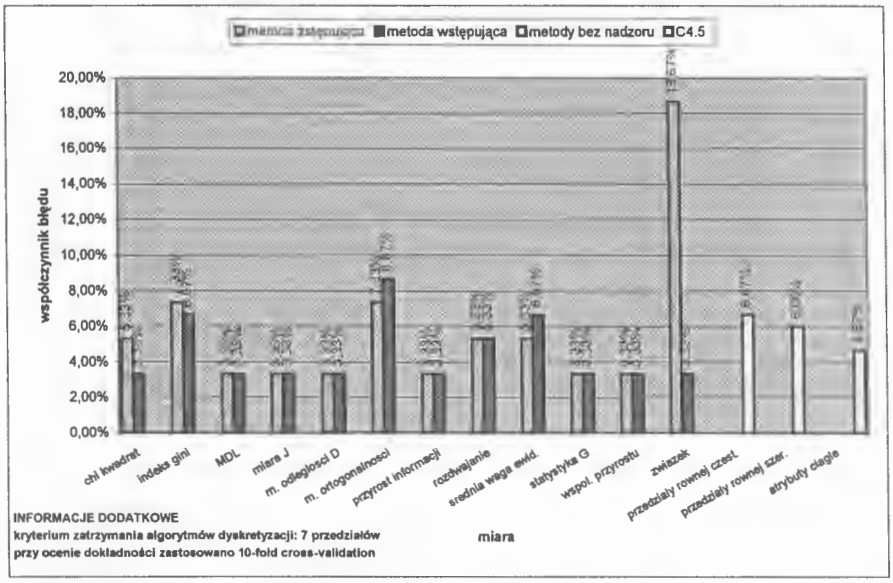
W drugiej części badań jakości drzew ponownie przyjęto, że każdy z atrybutów musi przyjąć stałą liczbę przedziałów. Badania przeprowadzono dla wzrastającej liczby przedziałów dyskretyzacji. Wyniki mają na celu ukazanie jaka jest w przybliżeniu najwłaściwsza liczba przedziałów dyskretyzacji pod względem minimalizacji poziomu błędów. W badaniach wykorzystano zbiór *glass*, algorytmy dyskretyzacji oparte na metodzie *zstępującej* i *wstępującej* oraz mierze *chi kwadrat* (rysunek 12). W celu porównania wyników przeprowadzono również takie same badania dla metod dyskretyzacji bez nadzoru (rysunek 13) oraz zamieszczono wyniki dyskretyzacji algorytmu C4.5. Algorytmem generującym wszystkie drzewa był C4.5 z wyłączonym przycinaniem.



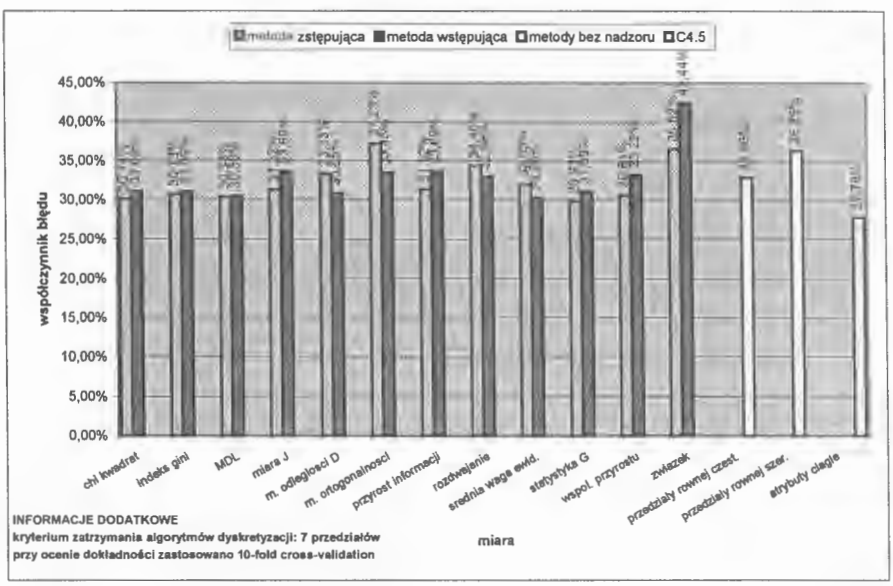
Rysunek 8. Współczynniki błędów dla algorytmu C4.5 (bez przycinania) dla zbioru danych diabetes zdyskretyzowanego przez opracowane algorytmy dyskretyzacji atrybutów ciągłych oraz dla zbioru z atrybutami ciągłymi



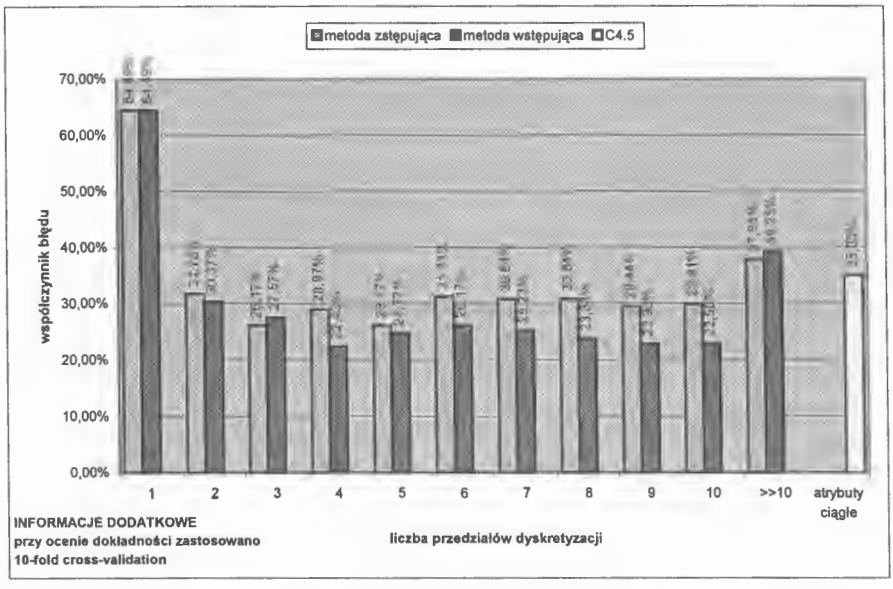
Rysunek 9. Współczynniki błędów dla algorytmu C4.5 (bez przycinania) dla zbioru danych glass zdyskretyzowanego przez opracowane algorytmy dyskretyzacji atrybutów ciągłych oraz dla zbioru z atrybutami ciągłymi



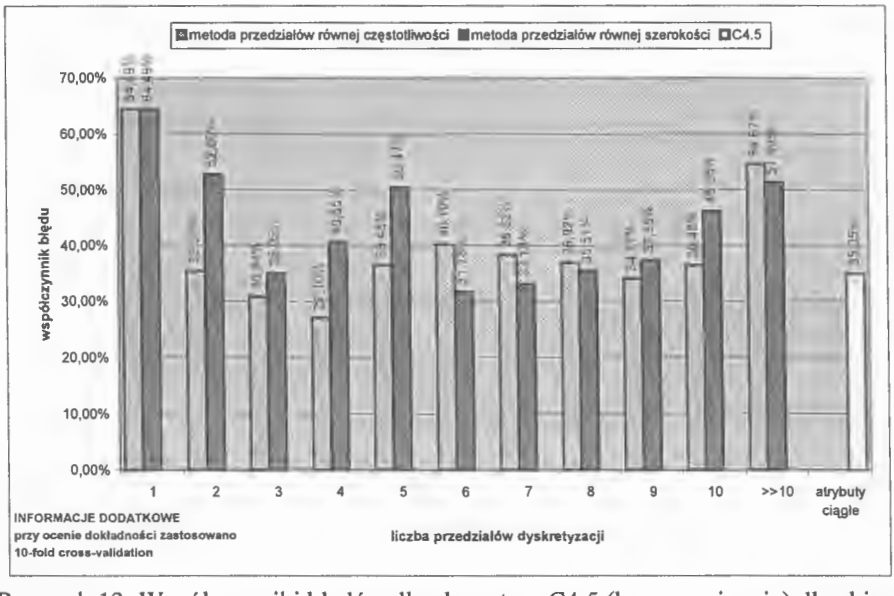
Rysunek 10. Współczynniki błędów dla algorytmu C4.5 (bez przycinania) dla zbioru danych iris zdyskretyzowanego przez opracowane algorytmy dyskretyzacji atrybutów ciągłych oraz dla zbioru z atrybutami ciągłymi



Rysunek 11. Współczynniki błędów dla algorytmu C4.5 (bez przycinania) dla zbioru danych vehicle zdyskretyzowanego przez opracowane algorytmy dyskretyzacji atrybutów ciągłych oraz dla zbioru z atrybutami ciągłymi



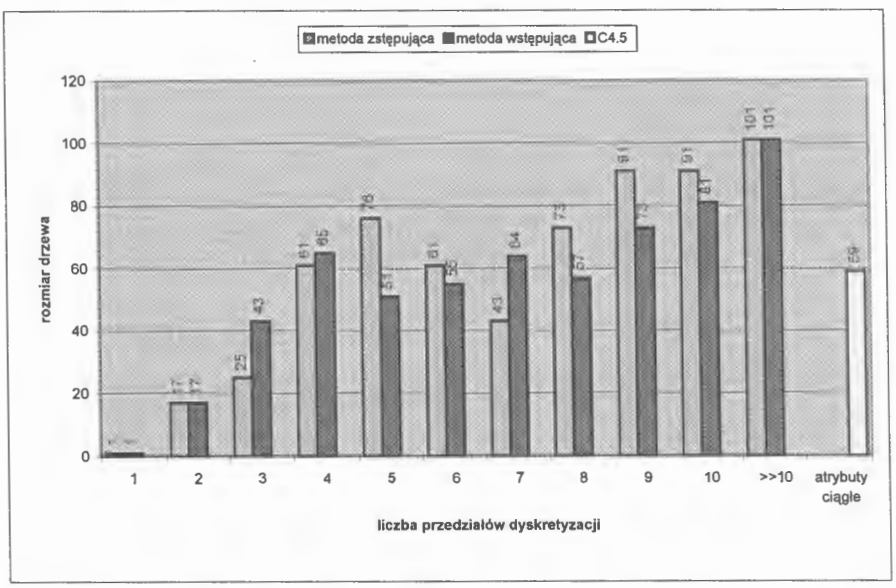
Rysunek 12. Współczynniki błędów dla algorytmu C4.5 (bez przycinania) dla zbioru danych glass zdyskretyzowanego przez opracowane algorytmy oparte na mierze chi kwadrat oraz dla zbioru z atrybutami ciągłymi



Rysunek 13. Współczynniki błędów dla algorytmu C4.5 (bez przycinania) dla zbioru danych glass zdyskretyzowanego przez metodę przedziałów równej częstotliwości i równej szerokości oraz dla zbioru z atrybutami ciągłymi

Badania rozmiarów drzew decyzyjnych

Poniższe wyniki mają na celu ukazanie w jakim stopniu rozmiar drzewa zależy od liczby przedziałów dyskretyzacji oraz w jakim stopniu metoda przycinania może zredukować rozmiar drzewa (redukcja rozmiaru drzewa ma zasadniczy wpływ na zrozumienie struktury drzewa). W badaniach wykorzystano zbiór *glass*, algorytmy dyskretyzacji oparte na metodzie *zstępującej* i *wstępującej* oraz mierze *chi kwadrat*. Do badań wykorzystano algorytm indukcji drzew decyzyjnych C4.5 (rysunek 14) oraz tenże sam algorytm z wyłączonym przycinaniem drzew (rysunek 15). W celach porównawczych zamieszczono również rozmiary drzew wygenerowane przez C4.5 dla atrybutów ciągłych.

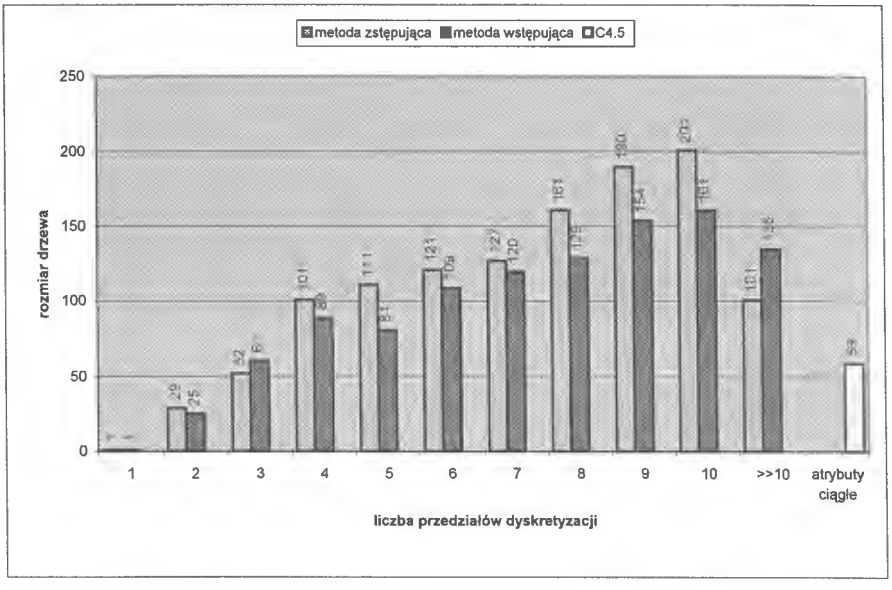


Rysunek 14. Rozmiary drzew uzyskanych dla algorytmu C4.5 dla zbioru danych glass zdyskretyzowanego przez opracowane algorytmy dyskretyzacji oparte na mierze chi kwadrat oraz dla zbioru z atrybutami ciągłymi

Badania czasu uczenia drzew decyzyjnych

Celem tego punktu jest ukazanie w jakim stopniu wzrost liczby przedziałów dyskretyzacji powoduje wzrost czasu uczenia (rysunek 16) oraz w jakim stopniu wzrost rozmiaru zbioru trenującego powoduje wzrost czasu uczenia (rysunek 17). Porównano wyniki czasu uczenia drzew dla atrybutów ciągłych oraz atrybutów dyskretnych. W badaniach wykorzystano algorytm dyskretyzacji oparty na metodzie *zstępującej* i mierze *chi kwadrat*. W badaniach ponownie zastosowano algorytm indukcji drzew decyzyjnych

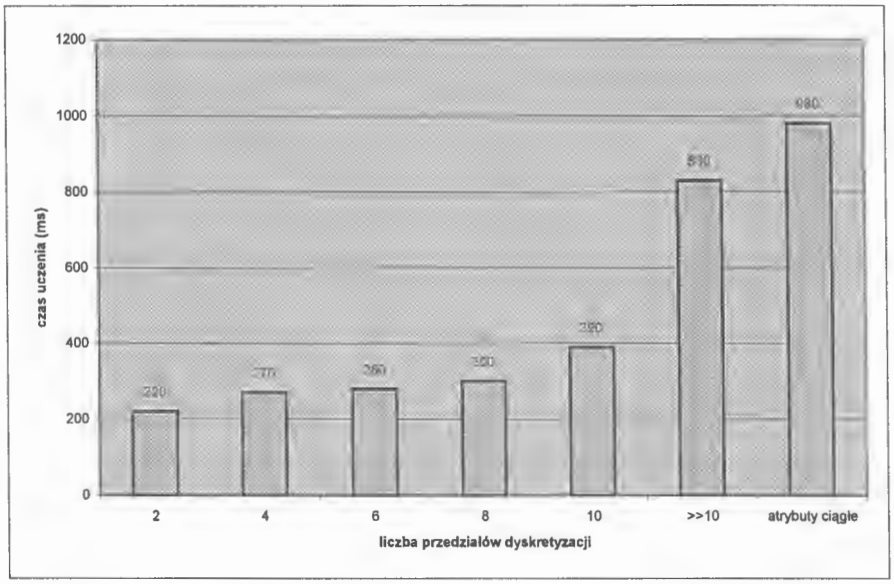
C4.5 z wyłączonym przycinaniem, gdyż algorytm przycinania zakłócałoby właściwy pomiar czasu uczenia, w pewnych przypadkach dokonując przycięcia w innych go nie dokonując. Badania wykonano na bazie zbioru *vehicle*. Wobec rysunku 17 istotna jest informacja, że *kryterium zatrzymania* algorytmu dyskretyzacji zostało ustawione na 5 przedziałów.



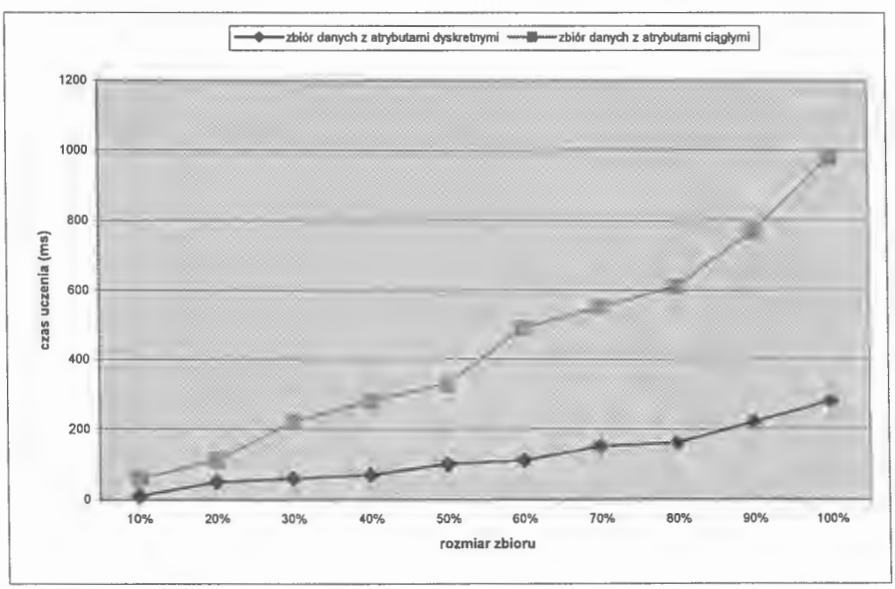
Rysunek 15. Rozmiary drzew (bez przycinania) uzyskanych dla algorytmu C4.5 dla zbioru danych glass zdyskretyzowanego przez opracowane algorytmy oparte na mierze chi kwadrat oraz dla zbioru z atrybutami ciągłymi

4. Wnioski

Przeprowadzone badania składają się z trzech części. Pierwsza część ma charakter odkrywczy, dotyczy oceny nowych algorytmów dyskrertyzacji pod względem jakości. Druga część ma charakter poglądowy, przedstawia pewne zależności dotyczące rozmiarów drzew decyzyjnych. Podobnie trzecia część prezentuje pewne aspekty dotyczące czasu uczenia drzew decyzyjnych. Celem wszystkich części jest zaprezentowanie wszystkich zalet płynących z dyskretyzacji wymienionych we wstępie niniejszej pracy. Oto kolejne punkty poruszające najważniejsze wyniki badań.



Rysunek 16. Czas uczenia dla algorytmu C4.5 (bez przycinania) przy użyciu całego zbioru danych vehicle dla różnej liczby przedziałów dyskretyzacji oraz dla atrybutów ciągłych



Rysunek 17. Czas uczenia dla algorytmu C4.5 (bez przycinania) dla różnych rozmiarów zbioru danych vehicle w wersji z atrybutami dyskretnymi oraz ciągłymi

Ocena jakości dyskretyzacji

Jakość dyskretyzacji uzyskanej przez stworzone algorytmy dyskretyzacji została oceniona poprzez ocenę jakości wygenerowanych drzew decyzyjnych. Celem badań przedstawionych na rysunkach 8, 9, 10, 11 było porównanie metody *zstępującej* ze *wstępującą* oraz porównanie *miar*. Jako punkt odniesienia posłużyły wyniki algorytmów dyskretyzacji bez nadzoru oraz wyniki dyskretyzacji dokonanej przez algorytm C4.5.

Na podstawie wspomnianych wykresów możemy stwierdzić *przewagę* algorytmów opartych na metodzie *zstępującej* w stosunku do algorytmów opartych na metodzie *wstępującej* pod względem jakości drzew decyzyjnych. Przykładowo dla zbioru danych *glass* (rysunek 9) przewaga jest bardzo wyraźna. Dla zbioru *diabetes* (rysunek 8) oraz *vehicle* (rysunek 11) przewaga jest również widoczna, chociaż trochę mniejsza niż w poprzednim przykładzie. Dla zbioru *iris* (rysunek 10) trudno stwierdzić przewagę którejkolwiek z metod.

Wobec przeprowadzonych badań zawartych w na rysunkach 8, 9, 10, 11 najlepsze wyniki ze względu na jakość drzew uzyskały algorytmy dyskretyzacji oparte na miarach: *statystyka G* (ang. G statistic) (Mingers 1987, 1989), *zasada minimalnej długości kodu* (ang. MDL - minimum description length) (Kononenko 1995), *chi kwadrat* (ang. chi square) (Agresti 1990). Przykładem najlepszych wyników uzyskanych dzięki *statystyce G* mogą być rezultaty dla zbioru *vehicle* (rysunek 11), kiedy to algorytm oparty na tej mierze i metodzie *zstępującej* przyczynił się do uzyskania najlepszego wyniku spośród opracowanych algorytmów. Podobnie algorytm oparty o miarę *chi kwadrat* i metodę *wstępującą* przyczynił się do uzyskania najlepszego wyniku dla zbioru danych *glass* (rysunek 9). Natomiast *MDL* uzyskiwał bardzo wyrównane i bardzo dobre wyniki dla wszystkich przetestowanych zbiorów danych.

Najgorsze wyniki uzyskały algorytmy oparte o *związek* (ang. relevance) (Baim 1988) oraz *miarę ortogonalności* (ang. orthogonality metric, angular disparity) (Fayyad i Irani 1992). Szczególnie algorytmy wykorzystujące *związek* bardzo często cechowały się bardzo dużymi wartościami współczynnika błędu i były dosyć nieprzewidywalnie. Przykładem tego mogą być wyniki dla zbioru *iris* (rysunek 10), gdzie *związek* dla metody *zstępującej* przyczynił się do otrzymania bardzo dużej wartości współczynnika błędu, natomiast dla metody *wstępującej* przyczynił się do otrzymania jednego z najlepszych wyników. Miara *ortogonalności* również wyróżniała się słabymi wynikami, co można zaobserwować na rysunkach 9, 10, 11.

Czasami różnice pomiędzy algorytmami opartymi o najlepsze i najgorsze miary były bardzo duże. Jednym z wielu przykładów świadczących o tym mogą być wyniki dla zbioru *glass* zamieszczone na rysunku 9, gdzie maksymalny współczynnik błędu dla algorytmu wykorzystującego miarę *związku* był ponad 2,5 razy większy od minimalnego współczynnika błędu dla algorytmu opartego na mierze *chi kwadrat*. Natomiast dla zbioru *iris* (rysunek 10) algorytm wykorzystujący *związek* przyczynił się do uzyskania błędu przeszło 5,5 razy większego od wyników uzyskanych przez algorytmy wykorzystujące najlepsze miary.

Interesujące jest to, że *statystyka G*, która jest jedną z miar opartych na przyroście informacji podobnie jak *współczynnik przyrostu* i *miara odległości D*, dla przyjętej konfiguracji przyczyniała się do osiągnięcia lepszych wyników (rysunki 8, 9, 11) od algorytmów opartych na *przyroście informacji* (również od algorytmu opartego na samym przyroście informacji). Tak więc miary mające na celu minimalizację preferencji atrybutów wielowartościowych, w przypadku atrybutów binarnych przyczyniają się do uzyskania odmiennej dyskretyzacji niż uzyskanej dla przyrostu informacji. Ciekawy jest również fakt słabych wyników algorytmów opartych na *mierze ortogonalności*, która została specjalnie stworzona dla atrybutów binarnych.

Wobec stworzonych algorytmów dyskretyzacji bardziej cennym punktem odniesienia, aniżeli wspomniane poniżej metody dyskretyzacji bez nadzoru były wyniki dyskretyzacji przeprowadzonej przez algorytm C4.5 na podstawie zbioru danych z atrybutami ciągłymi. Opracowane algorytmy dyskretyzacji oparte na lepszej część miar uzyskiwały zazwyczaj lepsze wyniki od algorytmu dyskretyzacji wbudowanego w C4.5. Przewaga tych algorytmów nie była tak duża jak w stosunku do algorytmów bez nadzoru, jednakże i tak dosyć wyraźna. Przykładem mogą być wyniki dla zbioru *glass* zawarte na rysunku 9, gdzie kilka opracowanych algorytmów dyskretyzacji uzyskało dużo lepsze wyniki od algorytmu dyskretyzacji zawartego w C4.5. Algorytmy te były oparte o miary *chi kwadrat*, *statystykę G*, *MDL*. Wyniki były lepsze o ok. 5-10% wartości współczynnika błędu. Podobnie, przewagę najlepszych algorytmów dyskretyzacji wobec dyskretyzacji przeprowadzonej przez C4.5 możemy zaobserwować dla zbiorów *diabetes* oraz *iris* (rysunki 8 i 10). Czasem dyskretyzacja algorytmu C4.5 również była bardzo dobra. Przykładem jest zbiór *vehicle*, dla którego algorytm dyskretyzacji zawarty w C4.5 uzyskał najlepsze wyniki w stosunku do wszystkich stworzonych algorytmów dyskretyzacji (rysunek 11).

Wyniki algorytmów dyskretyzacji *bez nadzoru* były zawsze gorsze od wyników najlepszych (wymienionych powyżej) algorytmów dyskretyzacji z

nadzorem. Zdarzało się również, że algorytmy dyskretyzacji bez nadzoru uzyskiwały dobre rezultaty, czego przykładem mogą być wyniki metody przedziałów równej szerokości, dla zbioru danych *glass*, ukazane na rysunku 9. Jednakże wyniki te są jedynie dziełem przypadku, gdyż algorytmy tego typu nie wykorzystują informacji o klasach (ang. class-blind).

Badania jakości drzew decyzyjnych zostały również przedstawione na rysunkach 12 i 13. Ich celem jest ukazanie zależności w jaki sposób zmienia się poziom błędów wraz ze zmianą liczby przedziałów dyskretyzacji oraz dla ilu przedziałów dyskretyzacji poziom błędów przyjmuje najniższe wartości. W tym celu wykorzystano zbiór *glass* oraz algorytm oparty o miarę *chi kwadrat* (rysunek 12). Badania powtórzono dla algorytmów dyskretyzacji *bez nadzoru* tj. metody przedziałów równej częstotliwości oraz równej szerokości (rysunek 13). Algorytm C4.5 posłużył do generacji drzew decyzyjnych.

Tak więc na podstawie rysunków 12 i 13 można zauważyć zależność polegającą na tym, że dla jednego przedziału dyskretyzacji dla każdego z atrybutów błąd osiąga dużą wartość. Dla kolejnych liczb przedziałów błąd maleje aż do osiągnięcia wartości minimalnej, a następnie znowu rośnie aż do maksymalnej liczby przedziałów (wówczas drzewo jest bardzo duże). *Minimalny* poziom błędów możemy zaobserwować dla 4-6 przedziałów dyskretyzacji zarówno dla algorytmu dyskretyzacji z nadzorem (rysunek 12) jak i dla algorytmów dyskretyzacji bez nadzoru (rysunek 13).

Dyskretyzacja 1-przedziałowa oznacza, że przykłady są od siebie nierozróżnialne. Pokazanie wyników dla 1 przedziału jest swego rodzaju punktem odniesienia – najgorszym akceptowalnym wynikiem, dla którego algorytm uczący się może nauczyć się jedynie tego, która z klas jest najliczniejsza (*współczynnik bazowy* – ang. base rate). Przeciwną skrajnością jest dyskretyzacja polegająca na generacji maksymalnej liczby przedziałów. Jeżeli zdarzy się, że w każdym z przedziałów znajdzie się jeden przykład, wówczas oznacza to, że brak jest jakiejkolwiek dyskretyzacji i algorytm uczący się nie jest w stanie się czegokolwiek nauczyć. Wtedy uzyskujemy najgorszy możliwy wynik w procesie uczenia. Na rysunkach 12 i 13 nie mamy takiego przypadku, lecz „dużą" liczbę przedziałów dyskretyzacji oznaczoną jako „>>10".

Na podstawie szerszych badań (nie zaprezentowanych w niniejszym opracowaniu) przeprowadzonych również dla innych zbiorów danych uzyskano podobne rezultaty co do liczby przedziałów, dla których poziom błędów był najniższy. Bardzo dobre wyniki uzyskano już dla 4 przedziałów dyskretyzacji. Ogólny wniosek jest taki, że warto silnie ograniczać liczbę

przedziałów dyskretyzacji. Dzięki temu drzewa są mniejsze, czytelniejsze, cechują się niższym współczynnikiem błędu oraz czas ich uczenia jest krótki.

Ocena rozmiarów drzew decyzyjnych

Na rysunkach 14 i 15 widzimy jak zmienia się rozmiar drzew wraz ze wzrostem liczby przedziałów dyskretyzacji oraz w jakim stopniu przycinanie drzew może zmniejszyć ich rozmiar. W celach porównawczych prezentujemy rozmiary drzew wygenerowanych dla atrybutów ciągłych. W badaniach wykorzystano zbiór *glass*, algorytm dyskretyzacji wykorzystujący miarę *chi kwadrat* oraz algorytm C4.5 w celu generacji drzew decyzyjnych.

Na omawianych rysunkach rozmiary nie przyciętych drzew są w przybliżeniu dwa razy większe aniżeli rozmiary drzew przyciętych i wzrastają wraz ze zwiększaniem się liczby przedziałów dyskretyzacji. Drzewa dla atrybutów ciągłych są dosyć małe. Osiągają rozmiary odpowiadające w bardzo zgrubnym przybliżeniu rozmiarom drzew dla 3 przedziałów dyskretyzacji (na obu rysunkach). Drzewa dla 1 przedziału dyskretyzacji składają się dokładnie z jednego liścia. Dla 1 przedziału dyskretyzacji dla każdego atrybutu przykłady są nierozróżnialne, wówczas zdegenerowane drzewa z jednym liściem „głosują" jedynie za najliczniejszą klasą i charakteryzują się wysokim współczynnikiem błędu. Natomiast dla dużej liczby przedziałów „>>10" drzewa są bardzo duże i cechują się również wysokim współczynnikiem błędu (rysunki 12, 13). Tak więc aby drzewo cechowało się najniższym współczynnikiem błędu nie może być zbyt małe ani zbyt duże. Warto stosować przycinanie, które pomaga w uniezależnieniu się od szumu i tym samym polepszeniu jakości generalizacji, a także w poprawie czytelności drzew.

Ocena czasu uczenia drzew decyzyjnych

Czas uczenia jest głównym powodem stosowania dyskretyzacji. Na rysunku 16 przedstawiamy jak zmienia się czas uczenia przy wzroście liczby przedziałów dyskretyzacji w stosunku do czasu uczenia dla atrybutów ciągłych. Natomiast na rysunku 17 ukazujemy jak zmienia się czas uczenia wraz ze wzrostem rozmiaru zbioru trenującego dla atrybutów ciągłych i dyskretnych. W badaniach wykorzystujemy zbiór *vehicle*, algorytm dyskretyzacji oparty o metodę *zstępującą* i miarę *chi kwadrat*. *Kryterium zatrzymania* stanowiło 5 przedziałów dyskretyzacji – uwaga istotna wobec wyników z rysunku 17.

Wraz ze wzrostem liczby przedziałów dyskretyzacji rośnie czas uczenia (rysunek 16). Jednakże, chociaż drzewo zbudowane dla dużej liczby

przedziałów dyskretyzacji osiąga rozmiary wielokrotnie większe od rozmiaru drzewa zbudowanego na atrybutach ciągłych to i tak czas jego budowy jest krótszy (rysunki 15 i 16). Przyczyną tego jest wspominana już konieczność czasochłonnego sortowania wartości każdego atrybutu ciągłego w procesie budowy drzewa. Wiedząc, że bardzo dobrą jakość drzew możemy osiągnąć dla 4-6 przedziałów dyskretyzacji, oszczędzamy na czasie w tym przypadku prawie 4-krotnie.

Wyniki zaprezentowane na rysunku 17 również w dużym stopniu zachęcają do stosowania dyskretyzacji, gdyż czas uczenia dla zbioru zdyskretyzowanego jest dużo mniejszy aniżeli dla zbioru z atrybutami ciągłymi.

5. Przyszłe badania

W niniejszym artykule przyjęto ustaloną liczbę przedziałów dyskretyzacji dla każdego atrybutu. Takie podejście służyło jedynie przyjętemu porównaniu przedstawionych algorytmów. W celu uzyskania najlepszych wyników dyskretyzacji należy zastosować kryterium zatrzymania stanowiące próg wartości miary z silnym ograniczeniem na liczbę przedziałów. Dobrym rozwiązaniem jest również zastosowanie *miary* np. *zasady minimalnej długości kodu*, czego przykładem może być metoda zstępującej minimalizacji entropii (Fayyad i Irani 1993). Przyjęcie dyskretyzacji *statycznej*, *globalnej* oraz podziału *binarnego* w opracowanych algorytmach służy przyspieszeniu procesu dyskretyzacji. Najlepszych wyników należałoby się jednak spodziewać po algorytmach dyskretyzacji *dynamicznej* pozwalających na wykrycie korelacji pomiędzy atrybutami, jednakże przy znacznym wzroście kosztów obliczeniowych.

Literatura

Agresti (1990) Categorical data analysis. John Wiley & Sons, New York.

Amarel S (1968) On the representation of problems of reasoning about action. Machine Intelligence, vol. 3, Edinburgh University Press.

Baim PW (1988) A method for attribute selection in inductive learning systems. IEEE Trans. On PAMI, vol. 10, 888-896.

Breiman L, Friedman JH, Olshen RA, Stone CJ (1984) *Classification and Regression Trees*. Wadsworth International Group

Catlett J (1991) On Changing Continuous Attributes Into Ordered Discrete Attributes. *Proceedings of the European Working Session on Learning*, Kondratoff Y, ed. Springer-Verlag, Berlin, 164-178.

Clark P, Niblett T (1989) The CN2 induction algorithm. *Machine learning*, vol. 3, 261-283.

Dougherty J, Kohavi R, Sahami M (1995) Supervised and unsupervised discretization of continuous features. *Proceedings of the Twelfth International Conference on Machine Learning*, Morgan Kaufmann.

Fayyad U, Irani KB (1992) The attribute selection problem in decision tree generation. *Proceedings of Tenth National Conference on Artificial Intelligence*, MIT-Press, Cambridge, 104-110.

Fayyad U, Irani KB (1993) Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning. *Proceedings of the Thirteenth International Joint Conference on AI*, vol.2, 1022-1027.

Holte RC (1993) Very Simple Classification Rules Perform Well on Most Commonly Used Datasets. *Machine Learning*, vol. 11, 63-91.

Kerber R (1992) ChiMerge: Discretization of Numeric Attributes. *Proceedings - Tenth National Conference on AI*, 123-128.

Kononenko I (1995) On Biases in Estimating Multi-Valued Attributes. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, vol. 2, 1034-1041.

Li M, Vitanyi P (1993) *An introduction to Kolmogorov Complexity and its applications*. Springer Verlag.

Liu WZ, White AP (1994) The Importance of Attribute Selection Measures in Decision Tree Induction. *Machine Learning*, vol. 15, 25-41.

Lopez de Mantaras R (1989) ID3 Revisited: A distance-based criterion for attribute selection. *Methodologies for Intelligent systems*, vol. 4, 342-350.

Lopez de Mantaras R (1991) A Distance-Based Attribute Selection Measure for Decision Tree Induction. *Machine Learning*, vol. 6, 81-92.

Martin JK (1997) An Exact Probability Metric for Decision Tree Splitting and Stopping. *Machine Learning*, vol. 28, 257-291.

van de Merckt T (1993) Decision Trees in Numerical Attribute Spaces. *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, vol.2, 1016-1021.

Michalski RS (1983) A theory and methodology of inductive learning. *Machine Learning*, Michalski RS, Carbonell JG, Mitchell TM, eds. Tioga, Palo Alto.

Michie D (1989) Personal Models of Rationality. *Journal of Statistical Planning and Inference*, vol. 21.

Mingers J (1987) Expert systems – rule induction with statistical data. *Journal of the Operational Research Society*, vol. 38, 39-47.

Mingers J (1989) An empirical comparison of selection measures for decision-tree induction. *Machine Learning*, vol. 3, 319-342.

Quinlan JR (1986) Induction of Decision Trees. *Machine Learning*, vol. 1, 81-106.

Quinlan JR (1993) *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

Quinlan JR (1996) Bagging, Boosting, and C4.5. *Proceedings of the Thirteenth National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference*, vol. 1, 725-730.

Richeldi M, Rossotto M (1995) Class-Driven Statistical Discretization of Continuous Attributes (Extended Abstract). *ECML 95*, Iraklion, Greece, 335-338.

Rissanen JR (1983) A universal prior for integers and estimation by minimum description length. *Annals of Statistics*, vol. 11, 416-431.

Smyth P, Goodman RM (1990) Rule induction using information theory. *Knowledge Discovery in Databases*, Piatetsky-Shapiro G, Frawley W, eds. MIT Press.

White AP, Liu WZ (1994) Bias in Information-Based Measures in Decision Tree Induction. *Machine Learning*, vol. 15, 321-329.

www.cs.waikato.ac.nz – *Strona domowa Uniwersytetu Waikato w Hamilton w Nowej Zelandii.*

ISBN 83-85847-74-X