

AUTOMATYCZNA IDENTYFIKACJA WYBRANYCH SYMBOLI NOTACJI MUZYCZNEJ

Marcin Luckner

Wydział Matematyki i Nauk Informacyjnych, Politechnika Warszawska

1 WSTĘP

Praca dotyczy problemu rozpoznawania symboli graficznych, a konkretnie elementów notacji muzycznej. Temat jest o tyle ciekawy, że wymaga rozważenia szeregu aspektów, od sposobu przedstawienia symboli, po nauczenie komputera elastycznych metod klasyfikacji. Same symbole muzyczne są interesujące i trudne do klasyfikacji ze względu na silne zróżnicowanie wewnątrz klas, występowanie znacznych zakłóceń i odkształceń powstałych wskutek przekształceń afinicznych.

Celem pracy było osiągnięcie skuteczności rozpoznania zbliżonego do dokonanego przez człowieka. Ponadto czas działania metod nie powinien być zbyt długi. Należy stawiać tak wysokie wymagania, aby udowodnić, że mozolna praca może zostać wykonana przez maszynę, mimo iż w powszechnej opinii wymaga ludzkiej inteligencji.

Dalsza część wstępu zawiera opis symboli muzycznych oraz problemów związanych z ich rozpoznawaniem.

W części dotyczącej analizy danych znajdują się rozważania na temat sposobu reprezentacji wzorców, ze szczególnym uwzględnieniem wektora cech i momentów Zernike.

Część poświęcona metodyce zawiera wyszczególnienie wykorzystanych algorytmów.

Po prezentacji wyników testów następuje część zawierająca wnioski.

1.1 Opis partytury

Partytura zawiera zapis muzyczny. Jest podzielona na poszczególne grupy, które łączą zapis

dla grających równocześnie instrumentów. Zapis dla pojedynczego instrumentu jest zawarty na pięciolinii.

Linie pięciolinii numerowane są od dołu. Odległość między nimi, którą będziemy nazywali odstępem, jest stała. Jednak na tej samej partyturze, a nawet w tej samej grupie, mogą się znaleźć pięciolinie o różnych odstępach. Symbole muzyczne mogą znajdować się zarówno na pięciolinii (klucze, znaczna część nut i pauz), jak i poza nią (większość dynamiki). Położenie nut wynika z wysokości dźwięku i to ich pozycja wymusza ustawienie innych symboli, tak aby uniknąć kolizji. W normalnych warunkach, to znaczy, gdy nie wpływa na nie sąsiedztwo nut, pozostałe symbole mają ustalone pozycje. Przesuwanie ich w pionie nie zmienia sensu zapisu muzycznego. Zazwyczaj wielkość symboli umieszczonych na pięciolinii jest uzależniona od jej odstepu. Zdarzają się jednak godne potępienia przypadki, gdy wydawcy zaznaczają drugi głos mniejszymi symbolami. To prowadzi do sytuacji w której na jednej pięciolinii znajdują się symbole o diametralnie różnych rozmiarach.

Nie istnieje jeden wspólny font dla zapisu nutowego. Wprowadza to znaczne zróżnicowanie symboli, szczególnie pauz $1/4$. Co więcej, istnieją fonty mające naśladować pismo ręczne. To prowadzi do zastąpienia symboli muzycznych nieczytelnymi szlaczkami. O ile pewna niedbałość jest zrozumiała i wytłumaczalna przy zapisie ręcznym, to stosowanie jej w przypadku fontów jest wręcz zaprzeczeniem ich idei.

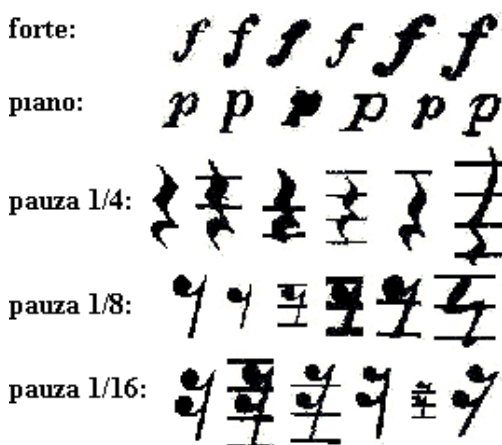
W pracy pomięto poważny problem lokalizacji symboli na partyturze, a jak łatwo się domyślić, nie jest to problem banalny. Rozpatrywane

przykłady pobrano ręcznie, wyznaczając prostokąt obejmujący wzorzec. Dbano przy tym, aby obiekt mieścił się całkowicie w opisującym go obszarze. Jeżeli sąsiedował z innymi obiektami, nie usuwano ich fragmentów z prostokąta.

Mimo pominięcia problemu lokalizacji klasyfikacja jawi się wyzwaniem. Mamy do czynienia z obiektami o zróżnicowanych wymiarach, zakłóconymi poprzez pięciolinie oraz sąsiadujące elementy notacji, a ponadto mocno zróżnicowane wewnątrz swych klas i przyjmujące czasem dość ekscentryczne formy.

1.2 Opis rozpoznawanych symboli

Rozpoznawano niektóre symbole dynamiki (piano, forte) i symbole reprezentujące pauzy (1/4, 1/8 i 1/16).



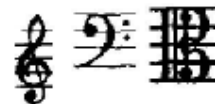
Rys. 1. Przykłady rozpoznawanych symboli.

Wybór tych elementów podyktowany był częstością ich występowania na partyturze oraz pewnym podobieństwem między nimi.

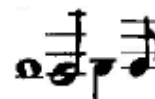
1.3 Inne symbole partytury

Należy zwrócić uwagę, że partytura zawiera wiele innych symboli i wskazane byłoby rozpatrzenie problemu błędnego rozpoznawania ich jako elementów jednej z pięciu klas podstawowych. Szczególnie należało by się przyjrzeć dwóm grupom symboli. Po pierwsze, symbolom najczęściej występującym na partyturze (nuty, klucze, znaki chromatyczne). Po drugie, symbolom podobnym do rozpoznawanych (inne pauzy i znaki dynamiki).

klucze:



nuty:



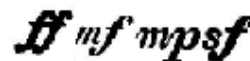
znaki chromatyczne:



pauza 1/32:



znaki dynamiki:



Rys. 2. Przykłady innych symboli z partytury.

2 ANALIZA DANYCH

Podstawowym problemem jest sposób, w jaki przedstawimy przypadki do klasyfikacji. Mamy przed sobą kilka trudności. Po pierwsze, przypadki mogą mieć różne rozmiary, czy to przez różne rozmiary czcionki, czy rozdzielczość skanowania. Przed procesem klasyfikacji należało by poddać je normalizacji. Po drugie, rozpatrywane przypadki są różnorodne, ze względu na mnogość czcionek. Po trzecie, są zakłócone przez pięciolinie, fragmenty sąsiednich elementów notacji muzycznej oraz proces skanowania.

Spróbujemy opisać przykłady za pomocą wektora cech, co będzie miało na celu wyeliminowanie przedstawionych problemów.

2.1 Normalizacja

Przed przystąpieniem do wyznaczania cech słuszne było by sprowadzenie wszystkich rozpatrywanych przypadków do jednakowej postaci, czyli dokonanie ich normalizacji. Niech docelowy rozmiar znormalizowanego obrazu wynosi 64x64. Musimy opisać rozpoznawane symbole okręgiem, a następnie przeskalować je tak, aby wpisać okrąg w docelowy prostokąt. Pozostaje pytanie, jak wyznaczyć środek takiego okręgu. Dobrym pomysłem wydaje się umieszczenie go w środku ciężkości obrazu. W ten sposób nawet jeżeli rozpatrywany obraz będzie przesunięty, wycentrujemy go.

2.2 Wektor cech

Bezpośrednie zapisanie obrazu o wymiarach 64x64 w wektorze o długości 4096 pociąga za

sobą nieprzyjemne konsekwencje, w szczególności co następuje:

- Rozmiar wektora danych wpływa znacząco na szybkość działania algorytmów.
- Brak odporności na przekształcenia afiniczne.
- Brak gwarancji wystąpienia cech wspólnych dla przypadków z tej samej klasy.

Powstaje zatem pytanie, jak opisać przypadki, zmniejszając wymiar przestrzeni danych oraz próbując wyeliminować pozostałe mankamenty. Rozwiązaniem może być wyodrębnienie cech opisujących przypadki, których wartości będą tworzyły nowy zestaw danych, nazywany wektorem cech.

Problem rozmiaru wektora cech możemy łatwo rozwiązać, narzucając maksymalną liczbę cech. Musimy przy tym pamiętać, aby nie ograniczyć go w stopniu, który uniemożliwi dokonanie klasyfikacji.

Kwestię przekształceń afinicznych częściowo rozwiązaliśmy w procesie normalizacji. Ustalając środek układu w punkcie ciężkości rysunku, zapewniamy niezmienność obrazu względem przesunięcia. Opisuując obiekty okręgiem, wpisywanym następnie w kwadrat o ustalonym rozmiarze, niwelujemy różnice wynikające ze skalowania. Niestety pozostaje problem obrotów. To zagadnienie da się jednak rozwiązać, stosując cechy będące niezmiennikami przekształceń afinicznych.

Problem występowania cech odróżniających poszczególne klasy jest w istocie sednem zadania klasyfikacyjnego i znalezienie ich stanie się *clou* rozważań.

Wydaje się, że ustalenie *a priori*, jakie cechy będą dobrze służyły naszemu procesowi klasyfikacji, jest niemożliwe. Pozostaje zatem eksperymentalny dobór cech. W tym celu zdefiniujemy zestaw cech, mogących reprezentować czarno-biały obraz o ustalonych rozmiarach 64x64, zapominając chwilowo o ocenie ich przydatności.

2.3 Wybrane cechy charakteryzujące obraz

2.3.1 Projekcje

Projekcją nazywamy rozkład opisujący liczbę czarnych punktów w poszczególnych kolumnach (projekcja wertykalna) lub w poszczególnych wierszach (projekcja horyzontalna). Każdą z projekcji można opisać wektorem długości 64 o wartościach z przedziału [0, 64].

2.3.2 Tranzycje

Tranzycje rozpatrujemy dla każdego wiersza (kolumny). Jest to liczba białych punktów poprzedzających bezpośrednio punkty czarne. Innymi słowy, jest to liczba zmian koloru w wierszu (kolumnie). Pamiętać będziemy je jako sumę tranzycji z poszczególnych wierszy i sumę tranzycji z poszczególnych kolumn. Tranzycje można przedstawić jako dwie wartości z przedziału [0, 2048].

2.3.3 Offset

Offsetem nazywamy liczbę białych punktów liczonych od brzegu obrazka do pierwszego czarnego punktu. Zapamiętamy margines lewy, prawy, górny i dolny. Aby ograniczyć wymiar wektora, będziemy pamiętali tylko najmniejszą wartość dla pasów szerokości ośmiu punktów. Można je opisać czterema wektorami długości 8 o wartościach z przedziału [0, 64].

2.3.4 Gęstość

Gęstością nazywamy stosunek liczby czarnych punktów do rozpatrywanej powierzchni. Rysunek dzielimy na szesnaście kwadratów i wliczamy gęstość dla każdego z nich. Gęstość rysunku opiszemy wektorem długości 16 o wartościach z przedziału [0, 100].

2.3.5 Skierowanie

Skierowanie określamy dla każdego czarnego punktu. Jest to kierunek najdłuższego odcinka złożonego z czarnych punktów i zawierającego rozpatrywany punkt. Rozpatrujemy odcinki o kącie nachylenia do poziomej krawędzi obrazu wynoszącym: 0, 45, 90, 135 stopni. Do opisu obrazu użyjemy wartości maksymalnych dla każdego z czterech kierunków: WE, NE, NS, NW (4 wartości z przedziału [0, 64]), oraz średnie wartości dla każdego kierunku w kwadratach 8x8 (cztery wektory długości 16 o wartościach z przedziału [0, 64]).

2.3.6 Gęstość skierowania

Gęstością skierowania nazywamy stosunek liczby czarnych punktów o zadanym skierowaniu do liczby wszystkich czarnych punktów. Liczymy ją dla pasów poziomych i pionowych o szerokości 8, dla każdego z kierunków. W sumie uzyskujemy cztery wektory długości 4 o wartościach z przedziału [0, 100] dla gęstości

wertykalnej i drugie tyle dla gęstości horyzontalnej.

2.4 Unifikacja wektora cech

Przy założeniu, że w procesie rozpoznania wykorzystamy wszystkie powyższe cechy, uzyskamy wektor długości 278. Jak wynika z opisu, mogą one przybierać wartości z różnych przedziałów. Aby w sztuczny sposób nie wywołać przez te różnice wrażenia większej istotności pewnych cech, powinniśmy dokonać ich unifikacji, tzn. sprowadzić do wspólnego przedziału. Łatwo to osiągniemy, stosując normę maksimum. Wartości każdej z cech dzielimy przez maksymalną wartość z przedziału, co sprowadza je do wartości z przedziału $[0, 1]$; dla naszej wygody przemnożymy te wartości przez 100 i zapamiętamy ich wartości całkowite. Teraz możemy porównywać poszczególne cechy.

2.5 Analiza danych

Zastanówmy się teraz nad przydatnością wybranych cech. Możemy przypuszczać, że nie wszystkie będą równie istotne. Jednak mając próbkę uczącą, możemy przeprowadzić ich analizę. Trzeba wziąć pod uwagę dwa aspekty. Wartości cech powinny być jak najbardziej zbliżone w poszczególnych grupach klasyfikacji i jak najbardziej zróżnicowane między nimi. Zazwyczaj do oceny zróżnicowania przestrzeni danych stosuje się algorytmy macierzowe, które rzutują wektory na przestrzeń o mniejszym wymiarze, modyfikując je zarazem tak, aby stosunek wyznaczników macierzy kowariancji wewnątrz klas i macierzy kowariancji między klasami był jak najmniejszy. Problem ten sprowadza się do rozwiązania zadania wektorów własnych (metoda PCA [5]) lub uogólnionej postaci tego zagadnienia (metoda MDA [6]). Niestety implementacja tych metod rodzi problemy, gdy rozpatrywane macierze są osobliwe. Zastosowano alternatywną metodę analizy wariacji cech.

2.5.1 Analiza wariacji

Bazując na metodzie MDA (ang. *Multiple Discriminant Analysis*) dobrano cechy do analizy tak, aby ich wariancja wewnątrz klas była minimalna, wariancja między klasami jak największa. Następnie posortowano cechy malejąco według stosunku wariacji między klasami do

wariancji wewnątrz klas i utworzono wektor o długości 47.

Istotność cech określono, zliczając wystąpienie cech w utworzonym wektorze i biorąc pod uwagę wagi będące odwrotnością długości wektora, do którego przynależą dane cechy.

Jeżeli pominiemy rozróżnienie kierunków, cechy, które mają największy wpływ na proces klasyfikacji, to: Offset, Skierowanie i Gęstość. Za nieprzydatne zostały uznane, i jako takie pominięte, cechy skalarne, tzn.: Tranzycje i Skierowanie Maksymalne. Dość nisko oceniane są Projekcje.

Rozpatrując cechy z uwzględnieniem kierunków, zauważymy zmiany w stosunku do poprzedniej analizy. Najistotniejsze składniki to: Gęstość skierowania (wertykalna) NS, Średnie skierowanie NW, NS, oraz Offset S. Istotnym elementem jest relatywnie niska istotność cech odpowiadających skierowaniu WE w stosunku do odpowiadających im cech o skierowaniu NS. Prawdopodobnie przyczyną tego braku jest występowanie pięciolini, skutecznie zakłócających adekwatne pomiary. Cechy dotyczące skierowania w kierunkach NW i NS powinny być odporne na te zakłócenia. Mniejsza istotność kierunku NE wydaje się wynikać z własności własnych sklasyfikowanych wzorców. Podobnie większe znaczenie Offsetu N i S w stosunku do kierunków W i E można przypisać zakłóceniu przez pięciolinię.

2.5.2 Analiza składników głównych (PCA)

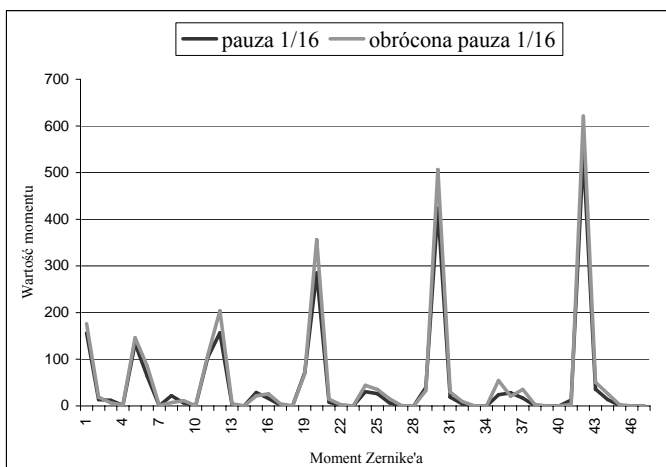
Metoda analizy składników głównych, w skrócie PCA (ang. *Principal Component Analysis*), pozwala osiągnąć dwa cele. Po pierwsze zmniejsza wymiar wektora, rzutując go na przestrzeń o mniejszym wymiarze. Po drugie rzutowanie odbywa się w sposób zapewniający obniżenie korelacji zbioru wektorów wejściowych, co powinno zwiększyć skuteczność uczenia. Niestety w procesie tym tracimy przejrzystość wyników, a nowe wektory są trudne do zinterpretowania. Metoda ta jest zaimplementowana w pakiecie *STATISTICA*.

2.6 Momenty Zernike'a

Aby zdefiniować momenty Zernike'a, należy powiedzieć parę słów o wielomianach Zernike'a. Są to zespolone, ortogonalne wielomiany, tworzące bazę funkcji całkowalnych po

powierzchni koła jednostkowego. Jako takie mogą służyć do przybliżenia funkcji spełniającej powyższy warunek, w szczególności do przybliżenia rozpatrywanych w tej pracy wzorców, za pomocą kombinacji liniowej o skończonej liczbie elementów. Współczynniki stojące przy jej elementach nazywamy momentami Zernike'a.

Momenty Zernike'a zostały po raz pierwszy zastosowane do analizy obrazu przez Teague [7]. Główna przyczyna użyteczności momentów Zernike'a to ich niezmienniczość względem przekształceń afinicznych, translacji, skalowania i obrotu.



Rys. 3. Zmiana momentów Zernike'a po obrocie o 30 stopni.

2.6.1 Zastosowanie momentów Zernike'a

Aby ocenić praktyczne zastosowanie momentów Zernike'a rozpatrzono przykład pauzy 1/16 obróconej o 30 stopni. Staranne skanowanie nie powinno raczej generować większych odchyżeń. Wyliczono 47 kolejnych wartości momentów Zernike'a dla wzorca i jego przekształcenia. Odległości między wektorami są nieznaczne, jak na rys. 3. Dlaczego jednak mamy w ogóle do czynienia z jakąkolwiek różnicą, skoro momenty te są niezmiennikami przekształceń afinicznych? Różnica wynika z błędów powstałych przy obliczaniu momentu Zernike'a. Błędy mają dwa źródła. Geometryczne, wynikające z aliasingu okręgu opisującego wzorec. Błąd ten jest warunkowany rozdzielczością wzorca. W naszym przypadku 64x64. Drugim źródłem są błędy numeryczne.

Różnica między momentami Zernike'a przed i po obrocie jest niewielka. Lecz jest to pojęcie względne. Potrzeba odnieść ją do przestrzeni danych. Mówiąc prościej, trzeba porównać ją

z odległościami, o których możemy wypowiedzieć się w sposób kategoriyczny. Na pewno można za duże przyjąć odległości między reprezentantami poszczególnych klas. Policzone zatem odległość między rozpatrywanym wzorcem, a przedstawicielami innych klas i porównano je z odległością od obrazu powstałego po obrocie. Różnica oddalenia od wzorca obróconego przykładu z tej samej klasy, a przypadków z innych klas, jest znaczna. Najbliższy obcy przykład jest położony około cztery razy dalej niż obraz obrócony.

3 METODY

Do rozpoznania wykorzystano sieci neuronowe i drzewa decyzyjne.

3.1 Sieci neuronowe

Charakter sieci neuronowych wydaje się odpowiadać rozpatrywanemu zadaniu. Sieci nie wymagają określenia ścisłych reguł klasyfikacyjnych, a dochodzą do nich samoczynnie na podstawie prezentowanej próbki uczącej. Ponadto nie ograniczają się do klasyfikacji zadanego zbioru, ale umożliwiają wykorzystanie nabytej wiedzy do oceny innych przypadków. W pracy wykorzystano sieci zaimplementowane w pakiecie *STATISTICA*:

- Perceptron wielowarstwowy (MLP).
- Liniowe sieci neuronowe.
- Radialne sieci neuronowe.
- Probabilistyczne sieci neuronowe (PNN).

Szczegóły dotyczące sieci można znaleźć w literaturze [1, 4]. Pojawiające się w pracy opisy perceptronu (np. MLP 33-18-6.) określają liczbę neuronów na poszczególnych warstwach.

3.2 Drzewa decyzyjne

W pracy wykorzystano zaimplementowane w pakiecie *STATISTICA* drzewa budowane algorytmem QUEST autorstwa Wei-Yin Loh i Yu-Shan Shih [2]. Służy on do tworzenia binarnych drzew decyzyjnych dla przypadków składających się zarówno z dyskretnych, jak i ciągłych cech. Istotną właściwością algorytmu jest możliwość klasyfikacji na podstawie kombinacji liniowej cech.

Zastosowane do opisu cechy potraktowano jako ciągłe i przeprowadzono badania dla

obydwu typów drzew, tzn. zarówno dla tych gdzie decyzja o podziale podejmowana jest na podstawie pojedynczej cechy, jak i tych, gdzie podstawą do decyzji jest kombinacja liniowa cech.

3.3 Problem generalizacji

Przytoczone metody można nauczyć dokładnego rozpoznawania przykładów z grupy uczącej. Jednak nas interesuje zdolność do rozpoznawania przykładów, które nie brały udziału w procesie nauki. Aby zapewnić zdolność generalizacji, należy podczas nauki metody badać jej skuteczność na grupie walidacyjnej, która nie bierze udziału w nauczaniu. Za końcową strukturę sieci lub drzewa uznajemy nie tę o najlepszych wynikach dla próbki uczącej, ale maksymalizującą wynik próbki walidacyjnej.

4 WYNIKI

Omawiany problem klasyfikacji polega na przypisaniu prezentowanych przypadków do jednej z pięciu wyszczególnionych klas. Problem klasyfikacji z odrzucaniem zakłada dodatkowo rozpatrywanie przypadków, które nie należą do żadnej z omawianych klas, i jako takie powinny zostać odrzucone.

Rozpatrywano 1500 przypadków, po 300 na klasę. Przykłady pochodzą z 90 partytur, liczących w sumie 107 stron. Wyszczególnienie liczby stron jest istotne, gdyż dla każdej z nich zniekształcenia powstałe w procesie skanowania mogą występować w innym natężeniu.

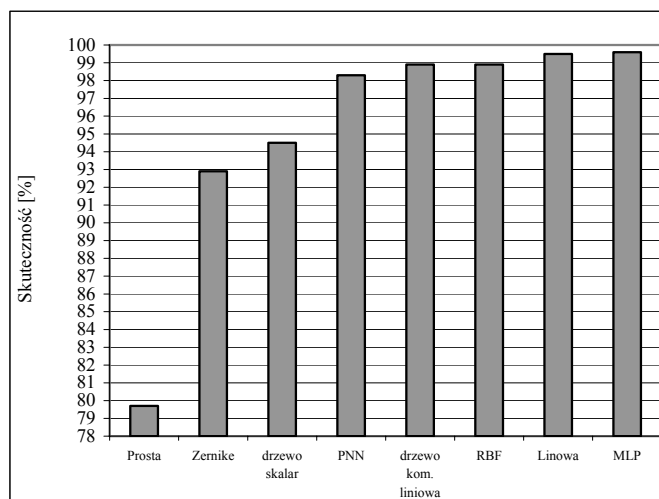
Przypadki podzielono na służące do nauki i testowe. Wyboru dokonano w sposób losowy co dało po 750 przypadków dla każdego z tych celów i około 150 na klasę. Przytoczone dalej wyniki dotyczą grupy testowej.

Dodatkowo na potrzeby klasyfikacji z odrzucaniem utworzono zbiór zawierający 149 przykładów spełniających jeden z warunków. Element należy do najczęściej występujących w notacji muzycznej albo jest w oczywisty sposób podobny do klasyfikowanych wzorców.

Tak samo jak dla innych klas zastosowano podział losowy na przykłady uczące (83) i testowe (66).

4.1 Klasyfikacja

Wyniki klasyfikacji pozwalają wyciągnąć interesujące wnioski. Najważniejszy dotyczy możliwości jej dokonania. Okazuje się, że zbiór 47 cech, wybranych w prosty sposób spośród 278 możliwych, jest w stanie zapewnić nam w pełni satysfakcjonującą klasyfikację. Jednak nie bez znaczenia pozostaje dobór metod. Zestawienie wyników przedstawia rys. 4.



Rys. 4. Wyniki klasyfikacji.

Na wstępie, aby określić, czy w ogóle należy stosować zaawansowane metody, wyliczono dla każdej klasy punkt będący wartością średnią jej reprezentantów w grupie uczącej. Przykłady klasyfikowano na podstawie odległości od tych punktów. Rezultaty tej prostej metody (ok. 80%) nie były imponujące. Jednak gdy tę samą technikę wykorzystano dla opisu momentami Zernike'a, skuteczność wzrosła do 93%. Należy tu zaznaczyć, że głównym źródłem błędów były przykłady mocno zaburzone przez pięciolinię. Dla przykładów spoza niej wyniki były znacznie lepsze. Okazuje się bowiem, że można dokonać klasyfikacji wszystkich przypadków nieleżących na pięciolinii, przy pomocy czterech momentów Zernike'a. Szczegóły można znaleźć w literaturze [3].

Uznając potrzebę wykorzystania bardziej zaawansowanych metod, przejdźmy do dalszej ich analizy.

Bardzo dobre wyniki możemy osiągnąć przy pomocy drzew decyzyjnych wykorzystujących algorytm QUEST i podział na podstawie kombinacji liniowej cech. Wykorzystanie tego samego

algorytmu przy podziale jedną cechą daje dalece mniej satysfakcjonujące rezultaty.

Praktycznie wszystkie sieci neuronowe mogą nas zadowolić, przy czym tylko MLP i sieć liniowa dają rezultaty znacznie lepsze od wyników drzewa. Należy tu wspomnieć o znacznie krótszym czasie nauki drzewa w porównaniu z każdym typem sieci. Także czas oceny przypadku, mimo wyliczania kombinacji liniowej, jest krótszy. W przypadku sieci o najszybszym czasie odpowiedzi, czyli sieci liniowej, działanie będzie ponad dwa razy dłuższe ze względu na strukturę drzewa, które wymaga obliczenia co najwyżej trzech kombinacji liniowych wektora cech. Szczególne znaczenie ma ten drugi przypadek. O ile proces uczenia można przeprowadzić wcześniej, to nawet niewielka różnica w czasie klasyfikacji przypadku odbije się znacznie na cierpliwości użytkownika chcącego przeanalizować stronę partytury.

4.2 Klasyfikacja z odrzucaniem

Niestety, stosując wektory cech, które sprawdziły się przy klasyfikacji, nie udało się uzyskać tak dobrych rezultatów dla klasyfikacji z odrzucaniem. Zarówno sieci neuronowe, jak i drzewa decyzyjne okazały się wrażliwe na dokonane zakłócenie w postaci wprowadzenia przykładów spoza rozpatrywanych klas. Co dziwne, drzewa decyzyjne z podziałem liniową kombinacją okazały się lepsze od większości sieci neuronowych. Jak pamiętamy, rozkład skuteczności w procesie klasyfikacji sugerował odwrotną relację.

Jedynym klasyfikatorem broniącym się w jakiś sposób jest sieć MLP 33-18-6. Jednak jej wynik (97.6%) jest o 2% niższy od odpowiednika zastosowanego w poprzednim procesie klasyfikacyjnym.

Można przejść nad pogorszeniem skuteczności do porządku dziennego, ale wyniki dla procesu klasyfikacji były tak obiecujące, że grzechem byłoby nie spróbować poprawić metod. Nasuwają się dwa sposoby niepociągające za sobą drastycznych zmian w danych i algorytmach. Możemy zwiększyć liczbę rozpatrywanych cech, wszak omawiane wektory początkowe miały wymiar 278. Ponadto możemy wybrać inne cechy do procesu klasyfikacji.

Pierwszy z tych wariantów zastosowano w przypadku sieci neuronowych, wykorzystując do nauki cały wektor cech. Drugą metodę

wykorzystano przy drzewach decyzyjnych. Zastosowano tu dwa warianty. W pierwszym zastąpiono najgorsze, w sensie analizy wariancji, cechy momentami Zernike'a. W drugim wykorzystano wektory powstałe w wyniku analizy składników głównych (PCA).

4.2.1 Drzewa decyzyjne

Rozpatrzmy klasyfikację opartą na wektorach cech długości 44, które składają się z momentów Zernike'a oraz najlepszych w sensie analizy wariancji cech z poprzednio rozpatrywanego wektora. Momenty Zernike'a stanowią ok. 50% uzyskanego w ten sposób wektora.

Wyniki wykazują poprawę w stosunku do klasyfikacji nieuwzględniającej momentów Zernike'a. Uzyskane 97.2% można porównać z wynikiem perceptronu wielowarstwowego, z poprzedniego paragrafu. Główną przyczyną błędu wydaje się odrzucanie prawidłowych wzorców, choć częściowo ma na niego wpływ błędna klasyfikacja elementów, które, przy pewnej dozie wyobraźni, mogą być uznane za podobne do przedstawicieli klasy (krzyżyki jako pauza 1/4).

Mimo podniesienia kosztu wyliczenia opisu wektora przez użycie momentów metoda ta jest znacznie szybszym klasyfikatorem niż klasyfikacja przy pomocy sieci. Przy tym podejściu możemy uznać zastosowanie momentów Zernike'a przy klasyfikacji drzewem decyzyjnym za pomysł udany.

Drugie podejście wymaga przekształcenia przestrzeni danych z 278 wymiarowej na 47 wymiarową przy użyciu analizy składników głównych. Sprowadza się to do przemnożenia wektora cech przez utworzoną wcześniej macierz. Wyniki (97.4%) są porównywalne z uzyskanymi przy pomocy momentów Zernike'a.

Minimalna poprawa wyniku nie jest wystarczającym argumentem dla stosowania tej metody. Pomijając potrzebę początkowego wyznaczenia macierzy, co nie jest zadaniem trywialnym, wykorzystanie jej w procesie wydłuża znacznie proces rozpoznania. Po pierwsze należy wyliczyć wszystkie 278 cech zamiast 47. Po drugie trzeba poddać je transformacji, przemnażając przez macierz.

Jak wynika z tej analizy, wydaje się sensowne zastosowanie wektora z momentami Zernike'a, co zwiększa skuteczność rozpoznania, nie pociągając za sobą zbyt wielkich kosztów. Wyniki są nadal o wiele gorsze od wyniku uzyskanego

w przypadku zastosowania drzew decyzyjnych w procesie klasyfikacji (98.9%).

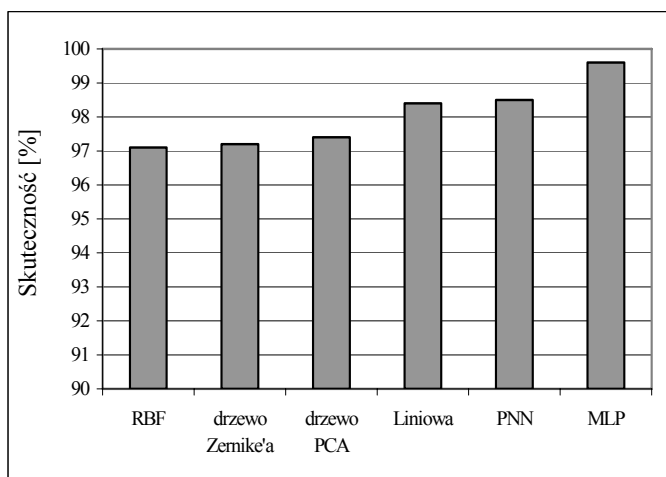
4.2.2 Sieci neuronowe

Sieć neuronowa nie daje możliwości tak łatwej analizy przydatności cech jak drzewa klasyfikacyjne. Zatem, wychodząc z założenia „więcej znaczy lepiej”, użyto jako wektora wejściowego wszystkich dostępnych cech. Od razu rzuca się w oczy skuteczność takiego postępowania. Aż trzy sieci uzyskują wyniki nie gorsze niż sieci neuronowe w podstawowym procesie klasyfikacji.

Co więcej, sieć o najwyższej skuteczności MLP 222-46-6 osiąga wynik 99.7%, co może już nas satysfakcjonować. Wyniki ukazują brak niekorzystnego wpływu, wynikającego z odrzucania innych elementów. Dla osłabienia naszego zadowolenia należy niestety zauważyć spadek szybkości rozpoznania, jaki wyniknie niewątpliwie ze znacznego wzrostu rozmiaru sieci.

4.2.3 Podsumowanie

Podjęcie próby poprawy skuteczności klasyfikacji z odrzucaniem zakończyło się sukcesem. Dotyczy to, choć w różnym stopniu, obu metod. Zestawienie na rys. 5.



Rys. 5. Wyniki klasyfikacji z odrzucaniem.

Niewielką, aczkolwiek wyraźną poprawę dało zastosowanie innego zbioru cech w przypadku drzew decyzyjnych. Odnosi się to zarówno do wykorzystania momentów Zernike'a, jak i analizy składników głównych. W tym drugim przypadku nakład kosztów obliczeniowych jest

jednak nieproporcjonalnie wielki w stosunku do stopnia poprawy.

Pełnym sukcesem zakończyła się natomiast próba poprawy wyników sieci neuronowych. Rezultaty trzech spośród nich są porównywalne z wynikami dla podstawowej klasyfikacji. Dzięki temu osiągnięciu zasłużyły na bliższą analizę.

Bezdiskusyjnym zwycięzcą, jest perceptron wielowarstwowy. Nie powinno nas to dziwić, gdyż ten rodzaj sieci uczony algorytmami wstecznej propagacji i metody gradientów odnosił sukcesy we wszystkich wcześniejszych rozważaniach. Zadziwiający jest jednak jego wysoki wynik (99.7%), który pozwala nam uznać problem klasyfikacji z obcymi wzorcami za rozwiązany. Niestety, następuje znaczne wydłużenie czasu klasyfikacji.

Ciekawostką jest nagły wzrost skuteczności sieci PNN. Ta skomplikowana sieć nie osiągała dotychczas rewelacyjnych rezultatów. Być może na jej skuteczność wpłynął znaczny rozmiar wektora, pozwalający wykorzystać opartą na rozkładzie mieszanym możliwość skupienia klasy wokół wielu ośrodków, co jest niewątpliwie przydatną cechą dla tak zróżnicowanej klasy jak badany zbiór „innych” przypadków.

Trzeci wynik uzyskała sieć liniowa. Szczerze mówiąc, budzi pewne zakłopotanie tak duża skuteczność najbardziej prymitywnej z omawianych metod. Mimo ostatniego miejsca spośród sieci o wynikach lepszych od wyników drzew, warto brać ją pod uwagę ze względu na szybkość działania. Dokonywana przez nią klasyfikacja jest najszybsza spośród rozpatrywanych sieci, choć nie tak szybka, jak w przypadku drzew.

5 WNIOSKI

Należy odpowiedzieć na pytanie, czy udało się osiągnąć zamierzone cele. Moim zdaniem tak. Skuteczność klasyfikacji w najlepszym przypadku wynosi 99.6%. W przypadku trudniejszym – klasyfikacji z odrzucaniem – osiągnąłem wynik 99.7%. Oznacza to odrzucenie 2-3 przykładów z populacji 750 przypadków (816 dla klasyfikacji z odrzucaniem). Ale założeniem pracy było sięgnąć szczytów, czyli dorównać rozpoznaniu dokonaneemu przez człowieka.

Nie wątpię, że nawet osoba nieobeznana z zapisem muzycznym, nie miałaby problemu z bezbłędnym rozpoznaniem każdego z klasyfiko-

wanych znaków. Oczywiście gdyby poproszono go o identyfikację pojedynczego przypadku. Jednak próba sklasyfikowania przedstawianych jeden po drugim 750 symboli nie byłaby raczej bezbłędna. Tu wkracza zmęczenie, znudzenie, omylność. Należy dodać, że klasyfikacja pojedynczego przypadku najwolniejszej z metod zajmuje ok. 0.15 sekundy.

Poza głównym celem pracy udało się także wykazać możliwość wykorzystania momentów Zernike'a dla klasyfikacji elementów niebędących zbyt mocno zakłóconymi (tzn. o „masie” zakłóceń mniejszej od „masy” symboli). Mam nadzieję, że da się tę właściwość wykorzystać w dalszym obalaniu dominacji ludzkiej inteligencji, przynajmniej w dziedzinie klasyfikacji.

BIBLIOGRAFIA

- 1) Bishop, Christopher M. 1997. *Neural Networks for Pattern Recognition*. Oxford University Press.
- 2) Loh, Wei-Yin. Shih, Yu-Shan. 1997. Split selection methods for classification trees. *Statistica Sinica*. 7, 815-840.
- 3) Luckner, Marcin. *Automatyczna identyfikacja wybranych symboli notacji muzycznej*. Praca magisterska. 2003.
- 4) Osowski, Stanisław. *Sieci neuronowe do przetwarzania informacji*. 2000. Oficyna Wydawnicza Politechniki Warszawskiej.
- 5) StatSoft. The *STATISTICA* Electronic Manual.
- 6) Sundaram, Ramasubramanian. *Multiple discriminant analysis*. Raport instytutowy. Department of Electrical and Computer Engineering Mississippi State University.
- 7) Teague, Michael Reed. Image analysis via the general theory of moments. 1980. *Journal of Optical Society of America*. 70, 8, 920-930.