STUDIA INFORMATICA 2010
Volume 31 Number 2A (89)

Rafał KUBIAK, Dominik NIEWIADOMY, Adam PELIKANT
Politechnika Łódzka, Instytut Mechatroniki i Systemów Informatycznych

AUTOMATYCZNE ZNAKOWANIE DANYCH AUDIO NA PLATFORMIE SERWERA BAZ DANYCH ORACLE

Streszczenie. W rozdziale tym przedstawiono projekt systemu automatycznego etykietowania nagrań dźwiękowych. System oparto na algorytmach nieliniowej transformacji czasu DTW, operującej na współczynnikach mel-cepstralnych i human-cepstralnych. Mechanizm automatycznego etykietowania korzystać będzie z w pełni konfigurowalnej, referencyjnej bazy nagrań oraz mapowań znaczników. Finalnie przestawione zostały testy potwierdzające wysoką jakość zaproponowanych algorytmów.

Słowa kluczowe: automatyczne znakowanie audio, tagi, MFCC, HFCC, DTW

AUTOMATED AUDIO DATA TAGGING SYSTEM FOR ORACLE DATABASE PLATFORM

Summary. In this chapter you will be provided with description of automated audio tagging system. The system will be based on optimized Dynamic Time Warping algorithm, mel-cepstral coefficients MFCC and human-cepstral coefficients HFCC. In addition the tagging process will be based on fully configurable reference audio database with mapping tags. Introduced tests results of proposed algorithms confirm their high-quality.

Keywords: automatic audio tagging, tags, MFCC, HFCC, DTW

1. Wprowadzenie

Współczesne systemy baz danych zawierające zbiory audio opierają swoje mechanizmy wyszukiwania na metadanych relacyjnych. Takimi metadanymi mogą być dane mówiące o autorze, informacje o albumie, z którego dane nagranie pochodzi bądź nazwie programu, z którego pochodzi ścieżka dźwiękowa. Innym potencjalnym podejściem możliwym do zaim-

plementowania w systemach bazodanowych są mechanizmy oparte na przeszukiwaniu strumieni dźwiękowych. Stosując algorytmy Query By Humming/Query By Voice jesteśmy w stanie wyszukać nagrania na podstawie fizycznego podobieństwa treści audio. Treść wyszukiwania może być podana na wejście systemu na kilka różnych sposobów, między innymi jako strumień audio, plik dźwiękowy, np. wave lub w przypadku utworów muzycznych jako zapis nutowy. Bardzo ciekawym zagadnieniem związanym z bazami audio jest automatyczne oznaczanie wybranej treści, tzw. tagowanie. W zależności od potrzeb nagrania audio trafiające do bazy dany za pomocą operacji INSERT i UPDATE mogą być sprawdzane pod kątem podobieństwa do zabronionej/znacznikowej treści. Niniejszy rozdział opisuje algorytmy zaprojektowanego systemu oznaczania nagrań audio zawierających predefiniowane wyrazy.

2. Analizowane zagadnienie

Zanim przystąpimy do zagadnienia projektowania jakiegokolwiek systemu informatycznego wykonujemy analizę jego użyteczności. Potrzeba implementacji języka zapytań SQL została uwarunkowana koniecznością pobierania informacji z relacyjnych baz danych. W zależności od konkretnego silnika bazodanowego, np. Oracle, powstały różne implementacje tego języka. Innym zagadnieniem praktycznym jest wyszukiwane danych z bazy zawierającej nagrania dźwiękowe (dalej nazywanej bazą audio). Powstały różne koncepcje implementacji tego zadania [1, 2, 3, 4]. Na szczególną uwagę zasługują algorytmy opartego o nucenie – Query by Humming bądź algorytmy oparte o zapytania głosowe – Query by Voice. Jednak mimo szerokiego spektrum odbiorców zapytania takie nie stały się do tej pory standardem.

Głównym aspektem związanym z bazami audio opisanym w tym rozdziale jest automatyczne oznaczanie plików zawierających specyficzną treść. Ręcznie nadawane znaczniki, czyli tzw. tagi stosowane są obecnie między innymi w serwisach www takich, jak chociażby blogi. Ich głównym zastosowaniem jest ułatwianie odszukiwania określonej zbliżonej tematycznie treści. Wykorzystanie algorytmów opisanych w niniejszym rozdziale pozwoli na automatyzację tagowania plików audio na podstawie zdefiniowanych fragmentów audio mapowanych na znaczniki. Innym przykładowym zastosowaniem tego typu algorytmów może być automatyczne oznaczanie treści niezgodnej z polityką serwisu (zakazane sentencje, słowa, wulgaryzmy).

3. Transformacja danych audio do MFCC i HFCC

Przetwarzanie danych audio, a w szczególności mowy jest zagadnieniem bardzo skomplikowanym. Złożoność ta wynika z faktu, że sygnał mowy:

- jest składową sygnałów pochodzących z otoczenia, a nie tylko od mówcy,
- jest złożeniem sygnału szumu oraz sygnału pochodzącego od wielu mówców,
- jest zależny od cech osobniczych i temporalnych mówcy,
- podlega ścieśnianiu oraz rozciąganiu w zależności od szybkości wypowiedzi.

Aby zminimalizować wpływ pierwszych trzech czynników na jakość klasyfikacji wprowadzone zostały algorytmy generujące współczynniki opisowe, bazujące na ludzkiej percepcji. Każda ramka przetwarzanego sygnału zostanie poddana transformacji do współczynników mel-cepstralnych MFCC (Mel Frequency Cepstral Coefficients) [5] oraz human-cepstralnych HFCC (Human Factor Cepstral Coefficients) [6].

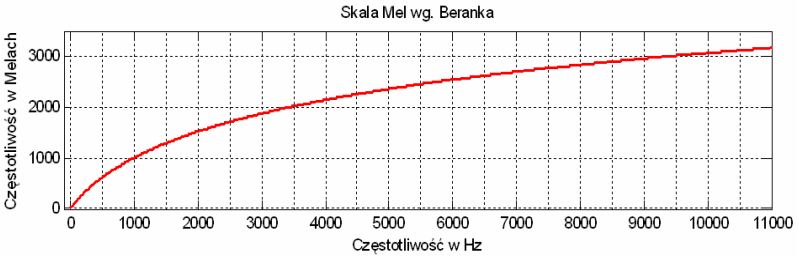
3.1. Generacja współczynników cepstralnych

Przy projektowaniu opisywanego systemu przyjęte zostało podejście perceptualne, które wskazuje, że najlepsze wyniki klasyfikacji uzyskuje się naśladując mechanizmy rozpoznawania sygnału mowy występujące u człowieka. Metodą pozwalającą dokonać tego jest przekształcenie częstotliwości tak, aby odpowiadała subiektywnemu odbiorowi przez ludzki słuch. W tym celu zastosowane zostały tzw. skale perceptualne, a w szczególności skala Mel. Zgodnie z definicją perceptualna skala Mel jest skalą dotyczącą wysokości tonu, czyli wrażenia słuchowego pozwalającego na określenie położenia tonu na skali częstotliwości. W wyniku wielu eksperymentów otrzymano następujący opis matematyczny transformacji skal:

$$f_{MEL} = 1127 \cdot \ln\left(1 + \frac{f_{Hz}}{0.7}\right) = 2595 \cdot \log_{10}\left(1 + \frac{f_{Hz}}{0.7}\right) \tag{1}$$

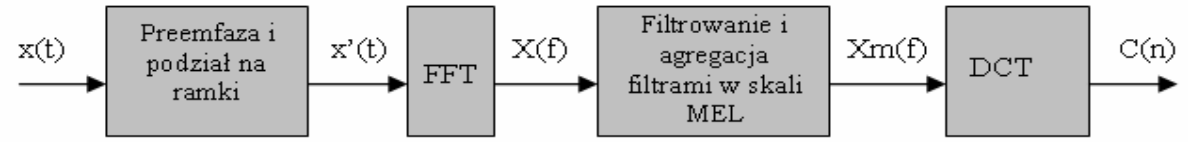
$$f_{Hz} = 700 \cdot \left[\exp\left(\frac{f_{MEL}}{1127}\right) - 1\right] \tag{2}$$

Poniżej przestawiona została zależność między częstotliwością liniową Hz, a częstotliwością perceptualną Mel.



Rys. 1. Skala perceptualna Mel
Fig. 1. Perceptual Mel scale

Na rysunku 1 przedstawiono nieliniowość skali. Można dostrzec, że skala Mel charakteryzuje się w przybliżeniu liniowym odwzorowaniem częstotliwości niskich oraz logarytmicznym częstotliwości wysokich. Implementacyjnie transformacja częstotliwościowa realizowana jest poprzez zastosowanie predefiniowanego banku filtrów, składającego się z rozłożonych równomiernie filtrów na skali perceptualnej, np. skali Mel. Opierając się na równaniach (1) i (2) w celu wygenerowania parametrów MFCC i HFCC należy wykonać kilkuetapowy proces przedstawiony na rysunku 2. Należy jednak dodać, że odpowiednio predefiniowany bank filtrów zachowuje swój kształt niezależnie od filtrowanego sygnału. Zatem przy zachowaniu warunków brzegowych (częstotliwości odcięcia i liczba filtrów) bank filtrów jest elementem niezmiennym w czasie.



Rys. 2. Ogólny algorytm generacji współczynników MFCC i HFCC
Fig. 2. MFCC and HFCC generation algorithm

Proces ten został już opisany w [5], jednak skrócony opis pomoże w zrozumieniu zagadnienia. Pierwszym krokiem przetwarzania sygnału jest filtracja oraz krótkookresowa transformata Fouriera (STFT). Następnie przy użyciu banku filtrów opisanego w dalszej części rozdziału (inny dla MFCC i HFCC) wykonywana jest filtracja, agregacja i logarytm przetworzonych danych. Ostatnim krokiem dla zadanej liczby współczynników jest wykonanie dyskretnej transformaty kosinusowej DCT i obliczenie parametrów pierwszej (Δ) i drugiej (ΔΔ) pochodnej współczynników.

3.2. Budowa banku filtrów dla współczynników HFCC

Algorytmy generowania współczynników MFCC i HFCC są algorytmami w pełni analogicznymi, z wyjątkiem jednego elementu, którym jest bank filtrów. Bank filtrów MFCC został opisany w [5], natomiast w celu wygenerowania współczynników Human Factor Cepstral Cofficients należy zbudować bank filtrów oparty na Equivalent Rectangular Bandwidth. ERB jest miarą w psychoakustyce, która aproksymuje pasma przenoszenia filtrów ludzkiego słuchu. W celu uproszczenia implementacji ERB przyjmuje założenie, że budowane filtry są prostokątnymi filtrami pasmowo przepuszczającymi. Na podstawie badań Moore'a i Glassberga wyprowadzona została następująca matematyczna forma opisu ERB:

$$ERB = \frac{\int |H(f)|^2 df}{|H(f_c)|^2} = af_c^2 + bf_c + c \tag{3}$$

$$a = 6.23 \cdot 10^{-6} \quad b = 93.39 \cdot 10^{-3} \quad c = 28.52 \tag{4}$$

Korzystając z zależności (3) oraz wiedząc, że bank filtrów w skali Mel jest złożony z filtrów trójKątnych, przechodzimy do następującego zapisu.

$$ERB = \int_{f_L}^{f_C} \frac{f - f_L}{f_C - f_L} df + \int_{f_C}^{f_H} \frac{f - f_H}{f_C - f_H} df = \frac{1}{2}(f_H - f_L) \tag{5}$$

Częstotliwość środkowa filtru w skali Mel można wyliczyć za pomocą równania (6). Korzystając z powyższych równań oraz z zależności między skalą perceptualną Mel a skalą liniową Hz możemy obliczyć środki filtrów dla wartości granicznych: f_{min} i f_{max}.

$$\hat{f}_{ci} = \frac{1}{2}\left(\hat{f}_{Hi} + \hat{f}_{Li}\right) \tag{6}$$

$$\log\left(1 + \frac{f_{ci}}{700}\right) = \frac{1}{2}\left[\log\left(1 + \frac{f_{Hi}}{700}\right) + \log\left(1 + \frac{f_{Li}}{700}\right)\right] \tag{7}$$

$$\log\left(1 + \frac{f_{ci}}{700}\right) = \frac{1}{2}\log\left[\left(1 + \frac{f_{Hi}}{700}\right)\left(1 + \frac{f_{Li}}{700}\right)\right] \tag{8}$$

Upraszczając to równanie otrzymujemy:

$$(f_{ci} + 700)^2 = (f_{Hi} + 700)(f_{Li} + 700), \tag{9}$$

co pozwala wyznaczyć następujące zależności:

$$f_{Hi} = \frac{f^2{}_{ci} + 700}{f_{Li} + 700} \quad f_{Li} = \frac{f^2{}_{ci} + 700}{f_{Hi} + 700} \tag{10}$$

Dla tak zdefiniowanych równań podstawiając odpowiednio f_{min} pod f_{Li} oraz f_{max} pod F_{Hi} otrzymujemy zależności dla krańcowych filtrów.

$$af_c^2 + bf_c + c = \hat{a}f_c^2 + \hat{b}f_c + \hat{c} \tag{11}$$

$$\hat{a}_1 = \frac{1}{2}\frac{1}{700 + f_{\min}} \quad \hat{b}_1 = \frac{700}{700 + f_{\min}} \quad \hat{c}_1 = -\frac{1}{2} f_{\min}\left(1 + \frac{700}{f_{\min} + 700}\right) \tag{12}$$

$$\hat{a}_M = -\frac{1}{2}\frac{1}{700 + f_{\max}} \quad \hat{b}_M = -\frac{700}{700 + f_{\max}} \quad \hat{c}_M = \frac{1}{2} f_{\max}\left(1 + \frac{700}{f_{\max} + 700}\right) \tag{13}$$

Upraszczając powyższy zapis otrzymujemy równanie:

$$f_{ci}^2 + \bar{b}f_{ci} + \bar{c} = 0 \quad \bar{b} = \frac{b - \hat{b}}{a - \hat{a}} \quad \bar{c} = \frac{c - \hat{c}}{a - \hat{a}}, \tag{14}$$

którego rozwiązanie pozwala obliczyć środki filtrów:

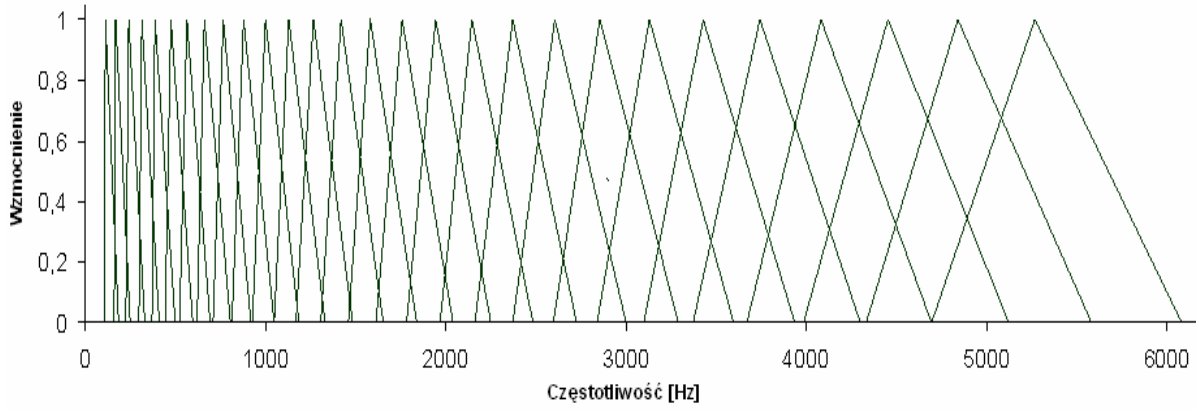
$$f_{ci} = \frac{-\bar{b} \pm \sqrt{b^2 - 4\bar{c}}}{2} \tag{15}$$

Wiedząc, że filtry w skali Mel mają wysokość równą 1, możemy obliczyć zależność między początkiem i końcem pasma przenoszenia. Dodatkowo, aby rozszerzyć szerokość filtrów wprowadzony został współczynnik ε, skalujący szerokość pasma przenoszenia.

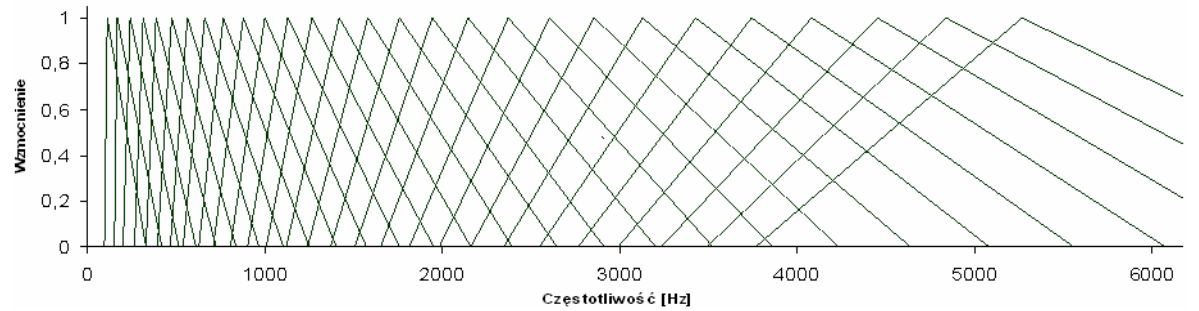
$$f_{Hi} = f_{Li} + 2 \cdot \varepsilon \cdot ERB_i \tag{16}$$

$$f_{Li} = -700 - \varepsilon \cdot ERB_i \pm \sqrt{(700 + \varepsilon \cdot ERB_i)^2 + f_{ci}{}^2 + 1400 f_{ci}} \tag{17}$$

Podstawiając wartości brzegowe do powyższych równań możliwa jest budowa banku filtrów. Poniżej przedstawione zostały 2 przykładowe banki filtrów różniące się parametrem ε.



Rys. 3. Bank 29 filtrów (f_{min}=20Hz, f_{max}=6kHz) ε=1
Fig. 3. 29 filters filterbank (f_{min}=20Hz, f_{max}=6kHz) ε=1



Rys. 4. Bank 29 filtrów (f_{min}=20Hz, f_{max}=6kHz) współczynnik ε=3
Fig. 4. 29 filters filterbank (f_{min}=20Hz, f_{max}=6kHz) ε=3

Posiadając bank filtrów możliwa jest generacja danych niezbędnych do dalszego przetwarzania. Ze względu na szczególny charakter sygnału mowy oraz wygenerowanych współczynników należy w kolejnym kroku wybrać klasyfikator, który będzie:

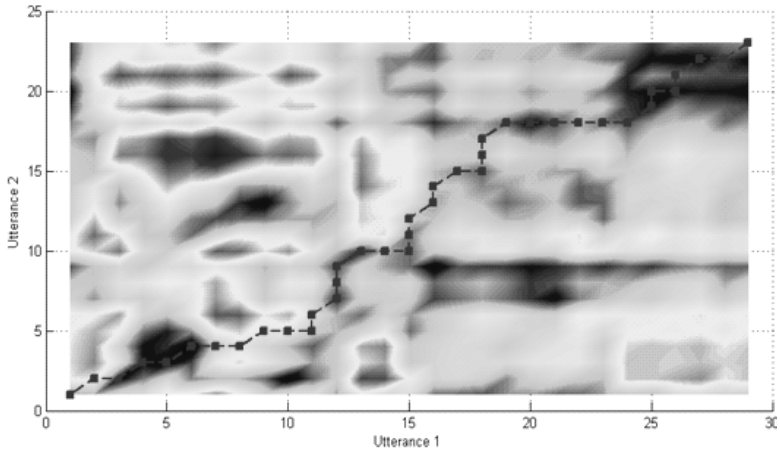
- w stanie pracować na 2 wymiarowych szeregach czasowych (ramki MFCC/HFCC),
- odporny na zmiany różną długość pasujących wzorców.

Klasyfikatorem spełniającym te wymagania jest klasyfikator nieliniowej transformacji czasu (Dynamic Time Warping).

4. Klasyfikator DTW

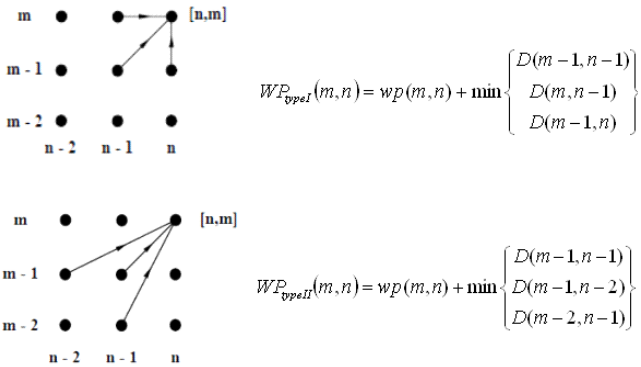
Klasyfikator nieliniowej transformacji czasu DTW jest mechanizmem sprawdzającym dopasowanie dwóch szeregów czasowych. Doskonale nadaje się do wyszukiwania wzorców dźwiękowych przy zachowaniu odporności na nierówną długość wypowiedzi. Dodatkowo algorytm ten poprzez odpowiednie zdefiniowanie metryk odległości pozwala na wykorzystywanie N-wymiarowych szeregów czasowych.

Klasyczna wersja algorytmu opiera swoje działanie na macierzy odległości każdej możliwej pary porównywanych wzorców. Do budowy macierzy odległości w ramach opisywanego systemu wykorzystywana jest metryka Euklidesowa. Poprawne dopasowanie dwu wzorców występuje, gdy monotoniczne niemalejąca ścieżka przejścia z punktu (0, 0) do punktu (M, N) jest zbliżona do przekątnej macierzy. Poniżej przedstawiony jest przykład dwóch dopasowanych wzorców z zaznaczeniem obliczonej ścieżki przejścia.



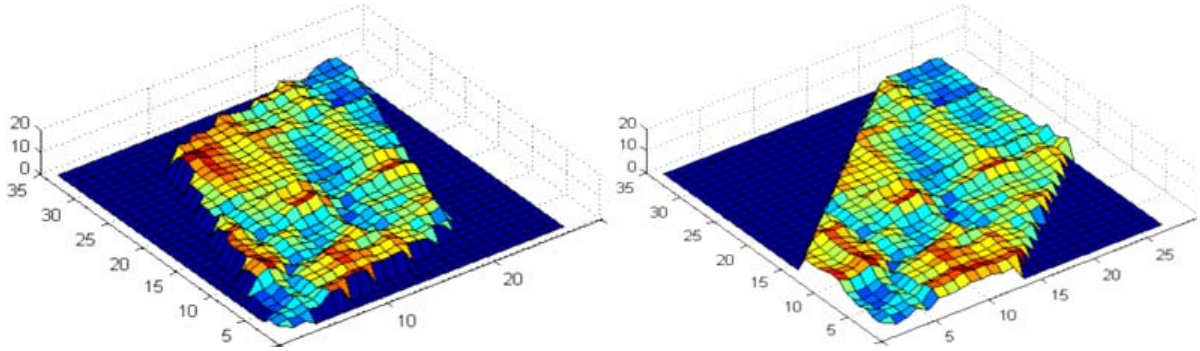
Rys. 5. Dopasowanie 2 wzorców z zaznaczeniem ścieżki przejścia
Fig. 5. DTW: 2 utterances matching with warping path

Dobór punktów ścieżki przejścia oparty jest na tzw. programowaniu dynamicznym korzystającym z definicji lokalnego ograniczenia (w tym szczególnym przypadku typ I). Poniżej przedstawiono definicję lokalnego ograniczenia typu I i II.



Rys. 6. Lokalne ograniczenia typu I i II
Fig. 6. Local constraint type I and type II

Wynikowa wartość dopasowania jest sumą wszystkich pośrednich kroków przejścia przez całą trasę. W celu optymalizacji obliczeniowej klasycznego algorytmu stosowane są ograniczenia globalne zdefiniowane poprzez zakresy, w których ścieżka przejścia nie może się znajdować. W opisywanym systemie dla klasycznego DTW przetestowane zostały dwa ograniczenia: równoległobok Itakura [7] oraz wstęga Sakoe-Chiba [8]. Rysunek 7 przedstawia macierz dopasowania z zaznaczonymi na niebiesko ograniczeniami.



Rys. 7. Równoległobok Itakura i wstęga Sakoe-Chiba
Fig. 7. Itakura parallelogram and Sakoe-Chiba band

Oprócz wielu zalet klasyczna wersja algorytmu DTW posiada duże ograniczenie dla opisywanego zagadnienia, gdyż pozwala jedynie na dopasowywanie wzorców jednostkowych. Ograniczenie to powoduje, że nie można skutecznie wyszukiwać podsekwencji, jeśli nie operujemy na nagraniach mowy izolowanej (nagrania mowy z wyraźną separacją słów). Aby móc zaimplementować mechanizm znacznikowania niezbędna jest funkcjonalność pozbawiona tego ograniczenia, co prowadzi do konieczności optymalizacji opisywanego algorytmu.

5. Zoptymalizowany klasyfikator DTW

Problem wyszukiwania podsekwencji w sekwencjach ściśle wiąże się z analizą strumieni i rozwiązaniem zaproponowanym przez zespół Y. Sakurai, C. Faloutsos, M. Yamamuro [9], nazwanym dalej algorytmem SPRING. Korzystając z sekwencji W i Z można zdefiniować problem wyszukiwania podsekwencji jako wyznaczenie takich podzbiorów Z, które są podobne w sensie dystansu DTW. Naiwna wersja algorytmu DTW polegałaby na przeszukaniu wszystkich możliwych podzbiorów sekwencji Z i zastosowaniu do każdego z nich osobnego wywołania DTW. W efekcie złożoność takiego algorytmu wynosiłaby $O(n^3m)$, przez co jego zastosowanie w takiej formie nie miałoby sensu implementacyjnego. Lepszym rozwiązaniem, lecz nadal niewystarczającym, byłoby wyznaczenie kwalifikujących się do porównania podzbiorów Z $[i_s: i_e]$ sekwencji Z ze złożonością O(n), a następnie porównanie ich ze wzorcem W. Taki algorytm ma złożoność o klasę niższą i wynosi $O(n^2*m)$, co nadal nie jest wynikiem

zadowalającym. Zaproponowana modyfikacja DTW, czyli algorytm SPRING, wprowadza dwa nowatorskie rozwiązania tego problemu: dodanie specjalnego elementu do sekwencji wzorca tzw. star-padding oraz zastosowanie pomocniczej macierzy dopasowań podsekwencji (STWM Subsequence Time Warping Matrix).

Prefiks specjalny, czyli tzw. star pudding, polega na uzupełnieniu sekwencji zapytania W dodatkową wartością, oznaczaną jako ”*”, dla której rezultat porównania z dowolnym elementem jest zawsze najlepszym możliwym do otrzymania wynikiem. Po tej zmianie algorytm opiera swoje działanie na wyszukiwaniu podobieństwa pomiędzy sekwencją W’ i Z. W efekcie nie trzeba już wyznaczać podsekwencji, a proces wyszukiwania dopasowań sekwencji W można przeprowadzić na pojedynczej macierzy porównań. Dzięki temu całkowita złożoność obliczeniowa maleje do O(n*m).

Wprowadzenie prefiksu pozwala ustalić, gdzie zakończyło się konkretne dopasowanie i jaką osiągnęło wartość. Niestety, nie umożliwia określenia, w którym miejscu rozpoczęła się ścieżka dopasowania, co jest równie ważną informacją. W tym celu została stworzona macierz dopasowań podsekwencji STWM (ang. Subsequence Time Warping Matrix). Optymalizacja ta polega na zmodyfikowaniu macierzy porównań tak, by każda komórka przechowywała informację o początku przechodzącej przez nią ścieżki dopasowania. Dodatkowo w związku z wyszukiwaniem podsekwencji zmianie musi ulec mechanizm ograniczeń globalnych. Wprowadzone zostało skalowalne pasmo, którym można określić maksymalne odchylenie długości odszukanego elementu od wzorca szukającego. Tabela 1 prezentuje uzupełnioną macierz STWM, gdzie:

- pogrubiona wartość to aktualna suma długości drogi,
- oznaczenie wskazujące kierunek przejścia (B – dół, BL – lewo i dołu, L – lewo),
- w nawiasie S – indeks startu, L – długość ścieżki,
- w nawiasie kwadratowym – położenie na ograniczeniu globalnym.

Tabela 1

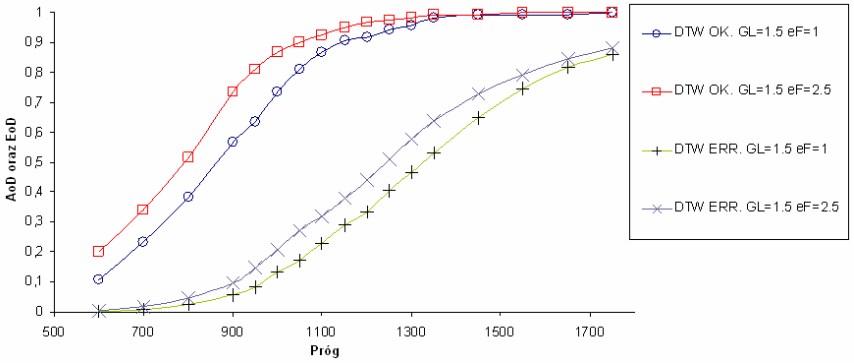
Wynik działania zoptymalizowanego DTW

REL	13,5	27,5	**3,5**	9,5	**3,5**	**4,5**	30,75
KOSZT:	54	110	**14**	38	**14**	**18**	123
W[4]=4	**54B** (S:1 ,L:4) [X: 1, Y: 4]	**110B** (S:2 ,L:4) [X: 1, Y: 4]	**14B** (S:2 ,L:4) [X: 2, Y: 4]	**38B** (S:2 ,L:4) [X: 3, Y: 4]	14B (S:4 ,L:4) [X: 2, Y: 4]	18B (S:4 ,L:4) [X: 3, Y: 4]	123B (S:5 ,L:4) [X: 4, Y: 4]
W[3]=9	53B (S:1 ,L:3) [X: 1, Y: 3]	**46B** (S:2 ,L:3) [X: 1, Y: 3]	**10B** (S:2 ,L:3) [X: 2, Y: 3]	**2BL** (S:2 ,L:3) [X: 3, Y: 3]	10B (S:4 ,L:3) [X: 2, Y: 3]	17BL (S:4 ,L:3) [X: 3, Y: 3]	42BL (S:5 ,L:3) [X: 4, Y: 3]
W[2]=6	**37B** (S:1 ,L:2) [X: 1, Y: 2]	**37B** (S:2 ,L:2) [X: 1, Y: 2]	**1BL** (S:2 ,L:2) [X: 2, Y: 2]	**17B** (S:4 ,L:2) [X: 1, Y: 2]	**1BL** (S:4 ,L:2) [X: 2, Y: 2]	26BL (S:5 ,L:5) [X: 3, Y: 2]	53B (S:7 ,L:2) [X: 1, Y: 2]
W[1]=11	**36B** (S:1 ,L:1) [X: 1, Y: 1]	**1B** (S:2 ,L:1) [X: 1, Y: 1]	**25B** (S:3 ,L:1) [X: 1, Y: 1]	**1B** (S:4 ,L:1) [X: 1, Y: 1]	**25B** (S:5 ,L:5) [X: 1, Y: 1]	36B (S:6 ,L:6) [X: 1, Y: 1]	4B (S:7 ,L:1) [X: 1, Y: 1]
	Z[1]=5	Z[2]=12	Z[3]=6	Z[4]=10	Z[5]=6	Z[6]=5	Z[7]=13

Tabela przedstawia wynik działania zoptymalizowanego DTW. Porównanie wzorca zapytania W=(11, 6, 9, 4) ze wzorcem znajdującym się w słowniku Z=(5, 12, 6, 10, 6, 5, 13) wskazuje, że najlepsze dopasowanie jest między indeksem Z [1] a Z [5].

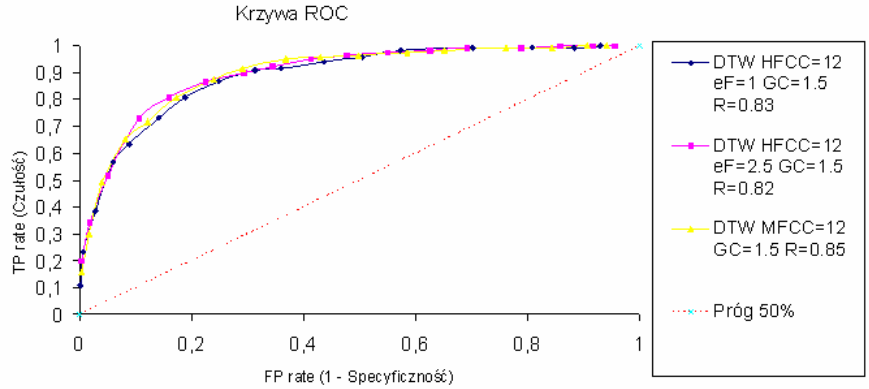
6. Wyniki i wnioski

W celu sprawdzenia jakości klasyfikatora dla opisywanych współczynników wykonanych zostało 1600 testów. Porównana została kombinacja każdy z każdym dla 40 nagrań mowy ludzkiej. Pozwoliło to określić skuteczność algorytmu w funkcji wybranych współczynników oraz progu klasyfikatora. Wynik zawiera procent poprawnie wyszukanych elementów (DTW OK) oraz procent wyszukanych, które nie powinny być wskazane (DTW ERR).



Rys. 8. Wyniki testów w funkcji progu
Fig. 8. Testing results In function of threshold

Dodatkowo jakość klasyfikatora dla danych wejściowych zweryfikowana została za pomocą krzywej ROC (Receiver Operating Characteristc) [10].



Rys. 9. Krzywa Reciver Operating Characteristic dla klasyfikatora
Fig. 9. Reciver Operating Characteristic curve for the classifier

Wyliczona wartość AUC, czyli pola pod powierzchnią krzywej ROC, wyniosła dla różnych kombinacji ustawień od 0,81 do 0,89, co stanowi o wysokim potencjale zastosowania badanego klasyfikatora.

Tak postawiona teza pozwala stwierdzić, iż opisane współczynniki oraz algorytm DTW pozwolą na implementację systemu automatycznego etykietowania plików audio. Zastosowanie klasyfikatora w formie wyzwalacza typu AFTER INSERT dla zdarzeń na kolumnie ze strumieniem dźwiękowym nie spowoduje nadmiernego obciążenia serwera bazy danych, a jednocześnie przyniesie wymierne korzyści dla systemu bazodanowego. Uniwersalność opisywanego zagadnienia pozwoli na różne formy implementacji akcji na zdarzenie wykrycia podobieństwa, począwszy od etykietowania, po blokadę treści lub przesłanie jej do moderacji. Akcja taka może zostać w pełni dostosowana do potrzeb konkretnego systemu i w prosty sposób będzie mogła być modyfikowana. Wszystkie te cechy wskazują na ewidentną potrzebę implementacji tak zaprojektowanego systemu.

BIBLIOGRAFIA

1. Weinstein E.: Query By Humming. A Survey, NYU and Google.
2. Ghias A., Logan J., Chamberlin D.: Query by humming – musical information retrieval in an audio database. ACM Multimedia 95, 1995.
3. Pelikant A., Niewiadomy D.: Klasyfikator podobieństwa w zapytaniach QBH oparty o współczynniki MFCC. BDAS, 2008.
4. Pelikant A., Niewiadomy D.: Query by Voice Example and sound similarity based on the Dynamic Time Warping algorithm. SMC, 2009.
5. Pelikant A., Niewiadomy D.: Implementation of MFCC vector generation in classification context. Journal of Applied Computer Science, 2008.
6. Skowronski M., Harris J.: Human Factor Cepstral Coefficients. J. Acoustical Society of America, Vol. 112, No. 5, Cancun, Mexico, Nov. 2002, s. 2305.
7. Sakoe H., Chiba S.: Dynamic programming algorithm optimization for spoken word recognition. Acoustics, Speech and Signal Processing, IEEE Transactions on, Vol. 26, No. 1.
8. Itakura F.: Minimum prediction residual principle applied to speech recognition. Acoustics Speech and Signal Processing, IEEE Transactions on, Vol. 23, No. 1, 1975, s. 67÷72.
9. Sakurai Y., Faloutsos C., Yamamuro M.: Stream Monitoring under the TimeWarping Distance. Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference.
10. http://en.wikipedia.org/wiki/Receiver_operating_characteristic .

Recenzenci: Dr inż. Jacek Frączek
Dr hab. inż. Ewa Piętka, prof. Pol. Śląskiej

Wpłynęło do Redakcji 31 stycznia 2010 r.

Abstract

In this chapter you will be provided with information how to implement algorithms that will enable you to create automated tagging system on Oracle Database platform. The main idea of such system is an automatic voice content tagging for specific utterances e.g. forbidden words. The presented implementation is based on audio perceptual Mel scale transformation done by Equivalent Rectangular Bandwidth filter bank and Human Factor Cepstral Coefficients. The process of HFCC generation is presented from scratch with ERB filter bank design. The chapter presents also how to implement and optimize Dynamic Time Warping algorithm by using SPRING modification proposed by Sakurai, Faloutsos and Yamamuro. Finally at the end of chapter the reader will be provided with digest of the classification results.

Adresy

Dominik NIEWIADOMY: Politechnika Łódzka, Instytut Mechatroniki i Systemów Informatycznych, ul. Stefanowskiego 18/22 90-924 Łódź, Polska, dominik.niewiadomy@gmail.com .
Adam PELIKANT: Politechnika Łódzka, Instytut Mechatroniki i Systemów Informatycznych, ul. Stefanowskiego 18/22 90-924 Łódź, Polska, apelikan@p.lodz.pl .