AGH University of Science and Technology

Seminarium – DSP AGH

Przegląd technik zwiększających wydajność obliczeniową algorytmów weryfikacji mówcy opartych o modelowanie GMM-UBM oraz HMM

Michał Salasa

06.03.2014, Kraków

Przedstawienie problemu

- **Systemy weryfikacji mówcy oparte o modelowanie GMM-UBM lub HMM do skutecznego działania wymagają stosunkowo dużej mocy obliczeniowej procesora.**

- **Duża ilość danych związana z ilością użytecznych parametrów wektora cech, długością weryfikowanej wypowiedzi i strukturą samego modelu GMM/HMM powoduje bardzo szybki przyrost obliczeń (przekleństwo wymiarowości).**

- **Komputery stacjonarne – mniejszy problem**

- **Systemy wbudowane – znaczący problem (SafeLock)**

Kategorie technik optymalizacyjnych

1) **Techniki ingerujące w przebieg procesu weryfikacji mówcy:**
   - modyfikacja sposobu obliczania logarytmu prawdopodobieństwa (PDE, BMP, DGS),
   - wykorzystywanie odpowiednio przekształconych modeli mówców (Sorted GMM).

2) **Techniki skutkujące zmniejszeniem ilości danych wykorzystywanych do weryfikacji:**
   - decymacja wektorów cech z pojedynczej wypowiedzi (FRD, VFR, ARD).

3) **Akceleracja sprzętowa i programowa:**
   - koprocesor zmiennoprzecinkowy - Cortex-M4F,
   - wykorzystywanie tablicowania (lookup table).

GMM-UBM, krótki opis metody

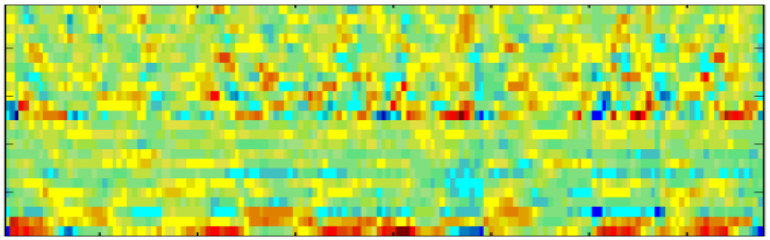
- **Model mówcy składa się z M mikstur (komponentów) i jest wytrenowany metodą MAP z modelu UBM**
- **Każdy komponent opisywany jest przez:**
  - wagę,
  - wektor wartości średnich (N-elementowy),
  - wektor wartości pochodzących z diagonalnej macierzy kowariancji (N-elementowy).
- **Weryfikacja polega na porównaniu wypowiedzi z modelem mówcy poprzez obliczenie logarytmu prawdopodobieństwa dla każdego wektora cech:**

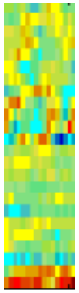
$$\log p(X \mid \lambda) = \sum_{t=1}^{T} \log p(x_t \mid \lambda) \qquad p(\boldsymbol{x} \mid \lambda) = \sum_{i=1}^{M} w_i \, p_i(\boldsymbol{x})$$

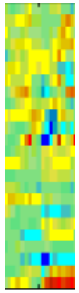
$$p_i(\boldsymbol{x}) = \frac{1}{(2\pi)^{D/2} |\Sigma_i|^{1/2}} \exp\left\{ -\frac{1}{2} (\boldsymbol{x} - \boldsymbol{\mu}_i)' (\Sigma_i)^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_i) \right\}$$

GMM-UBM, opis metody

- **Proces weryfikacji:**







$\log(p(\overrightarrow{x_t}|\lambda))$ + $\log(p(\overrightarrow{x_t}|\lambda))$ + $\log(p(\overrightarrow{x_t}|\lambda))$ + ... + $\log(p(\overrightarrow{x_t}|\lambda))$

t = 1 t = 2 t = 3 t = T

GMM-UBM, opis metody

- **Wynik końcowy:**

$$score = \sum_{1}^{T} \log(p(\vec{x}_t|\lambda) - \sum_{1}^{T} \log(p(\vec{x}_t|\lambda_{UBM})$$

- **T – normalizacja**: $score_T = \frac{score - \mu}{\sigma}$,

μ - średnia wyników dla wybranych modeli mówców (ok. 30)

σ – odchylenie standardowe wyników tych modeli

GMM-UBM, złożoność obliczeniowa

- **Czynniki, które w znaczący sposób wpływają na złożoność obliczeniową metody GMM-UBM to:**
  - wymiarowość wektora cech,
  - ilość wektorów cech (długość wypowiedzi),
  - struktura modelu mówcy (ilość komponentów),
  - ilość modeli wybranych do t-normalizacji,
  - potrzeba obliczania dużej ilości logarytmów.
- **Złożoność obliczeniowa GMM-UBM**: $O(M + M)$

GMM-UBM, wstępna optymalizacja

- **Obliczanie logarytmu prawdopodobieństwa dla pojedynczego komponentu:**

$$\log(p_i(\vec{x}|\lambda) =$$

$$= \log\left(\frac{\omega_i}{(2\pi)^{\frac{D}{2}}|\Sigma_i|^{\frac{1}{2}}}\exp\left\{-\frac{1}{2}(x-\mu_i)^{'(\Sigma_i)^{-1}}(x-\mu_i)\right\}\right) =$$

$$= \log\left(\frac{\omega_i}{(2\pi)^{\frac{D}{2}}|\Sigma_i|^{\frac{1}{2}}}\right) + \log\left(\exp\left\{-\frac{1}{2}(x-\mu_i)^{'(\Sigma_i)^{-1}}(x-\mu_i)\right\}\right) =$$

GMM-UBM, wstępna optymalizacja

$$= CONST + \left\{ -\frac{1}{2} \left(x - \mu_i \right)^{'(\Sigma_i)^{-1}} \left(x - \mu_i \right) \right\}$$

- **Dodanie do modelu mówcy dodatkowego wektora zawierającego stały czynnik normalizacyjny każdego komponentu, pozwala zaoszczędzić wiele mocy obliczeniowej (brak dużej ilości logarytmów) i przyspieszyć proces weryfikacji.**

GMM-UBM, Top-C mixtures

- **Najpopularniejszą techniką zwiększającą wydajność obliczeniową metody GMM-UBM jest technika zwana ‚top-C mixtures':**
  - Modele mówców muszą być wytrenowane metodą MAP
  - Wykorzystanie faktu, że jeden wektor cech jest modelowany niewielką ilością komponentów oraz, że komponenty UBM modelujące ten wektor w dużym stopniu pokrywają się z komponentami wytrenowanego modelu mówcy.
  - Ilość komponentów wykorzystywanych do obliczania końcowego wyniku jest redukowana do małej ilości (zazwyczaj C = 5)
  - Znaczne zmniejszenie ilości obliczeń kosztem niewielkiego spadku (a w niektórych przypadkach nawet wzrostu!) skuteczności weryfikacji

GMM-UBM, Top-C mixtures

- **Zmodyfikowany proces weryfikacji:**
    - Dla aktualnie badanego wektora cech należy obliczyć logarytm prawdopodobieństwa każdego komponentu modelu UBM.
    - Wybrać C komponentów cechujących się najlepszym wynikiem.
    - Obliczenia dla modelu weryfikowanego mówcy oraz modeli wybranych do t-normalizacji przeprowadzić już tylko dla tych C wybranych komponentów.
- **Uzyskana złożoność obliczeniowa:** O(M + C)

GMM-UBM, Sorted GMM

- **Kolejna technika – Sorted GMM – jest rozszerzeniem poprzedniej. Skupiono w niej uwagę na proces poszukiwania C-najlepszych komponentów modelu UBM.**
    - Aby zwiększyć efektywność poszukiwań komponenty modelu UBM są porządkowane według określonego klucza tworząc nową strukturę nazywaną: Sorted GMM
    - Porządkowanie (sortowanie) odbywa się względem zmiennej s(.), która jest funkcją parametrów komponentów.
    - Umiejętne wykorzystanie posortowanej struktury pozwala na szybsze znalezienie C-najlepszych komponentów

GMM-UBM, Sorted GMM

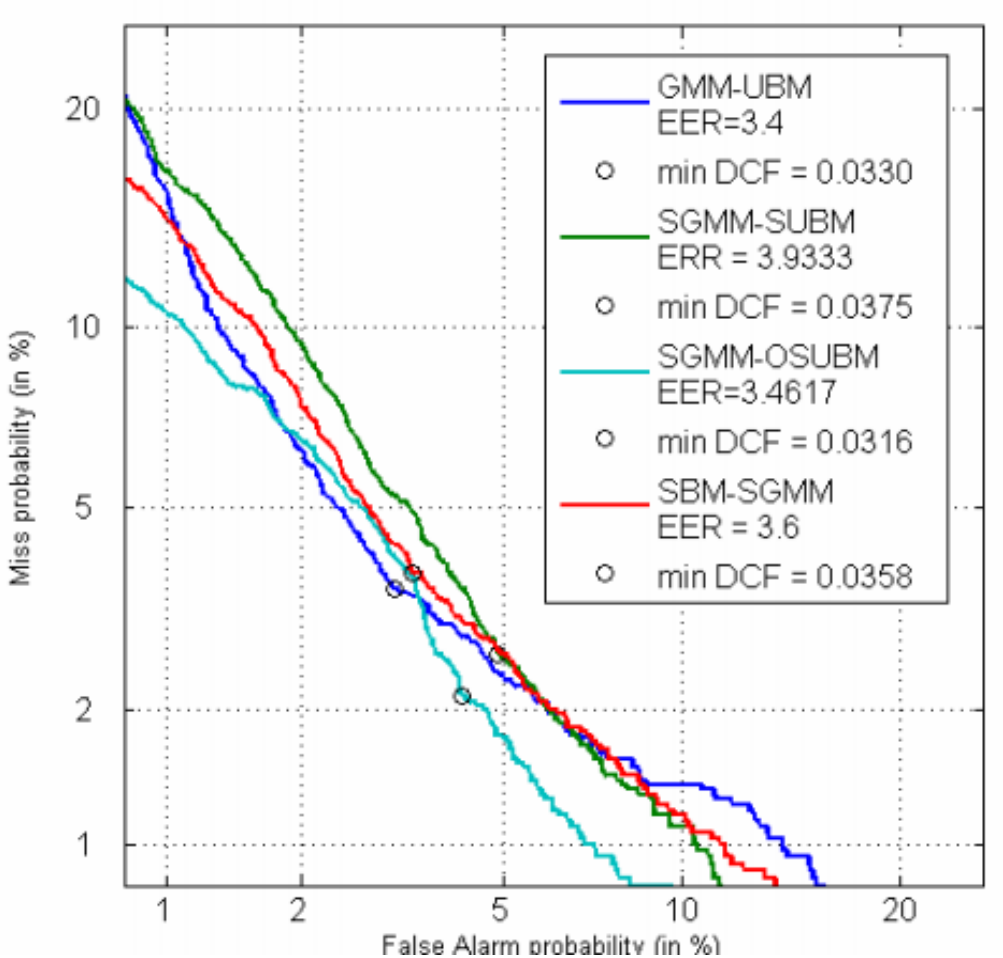
- **Przykładowe założenia procesu weryfikacji (H. Mohhamadi, R. Saeidi):**
    - Zmienna s(.) = suma wszystkich składników wektora wejściowego
    - Sorted GMM – mikstury posortowane względem sumy elementów wektora wartości średnich
    - M_s – ilość badanych komponentów (zawsze mniejsza do ilości komponentów w modelu)

GMM-UBM, Sorted GMM

- **Weryfikacja oparta o te założenia:**
  - Sortujemy odpowiednio komponenty modelu UBM otrzymując wektor $S = [s_1 + s_2 + \ldots + s_M]$
  - Dla badanego wektora cech obliczamy sumę jego elementów składowych: S_x
  - Dokonujemy kwantyzacji skalarnej sumy S_x względem wektora S otrzymując element s_i
  - Obliczamy logarytm prawdopodobieństwa tylko dla komponentów odległych o $M_s / 2$ od komponentu o indeksie i
  - Wybieramy C najlepszych komponentów

GMM-UBM, Sorted GMM

- **Złożoność obliczeniowa Sorted GMM:** $O(M_s + C)$



GMM-UBM, decymacja wektorów cech

1) Fixed-Rate Decimation

- Dla każdej wypowiedzi, do obliczania logarytmu prawdopodobieństwa brany jest co N-ty wektor cech
- Redukcja ilości wektorów cech na poziomie: 1/N

2) Variable Frame Rate

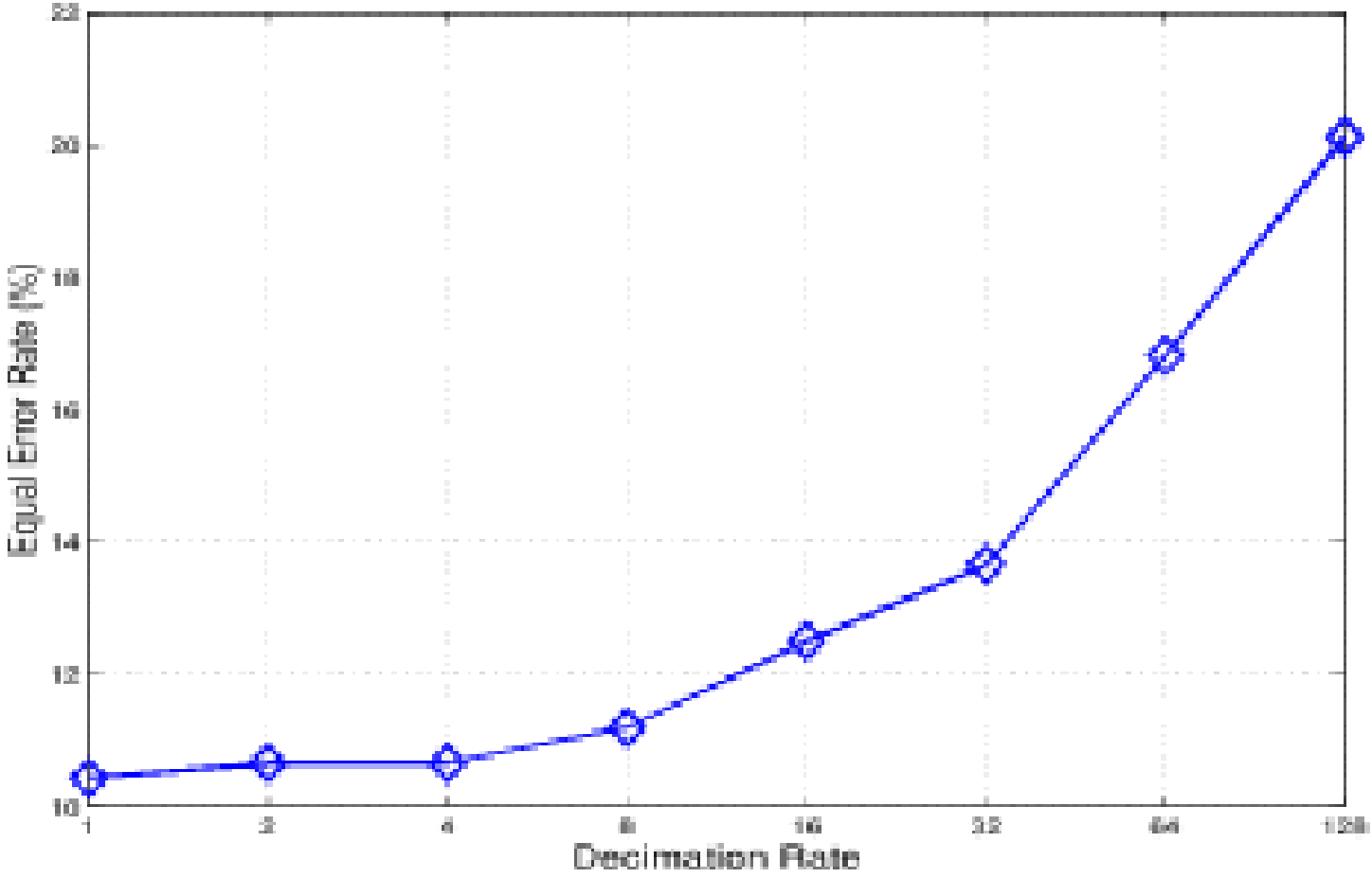
- Porównuje kolejne wektory cech i jeśli są do siebie dostatecznie podobne to logarytm prawdopodobieństwa obliczany jest tylko dla jednego z nich, a wynik dla pozostałych wektorów jest powielany
- Miara podobieństwa: najczęściej metryka euklidesowa

3) Adaptive-Rate Decimation

- Współczynnik decymacji jest adaptowany do bieżącej wypowiedzi tak aby zawsze otrzymać taką samą ilość wektorów cech

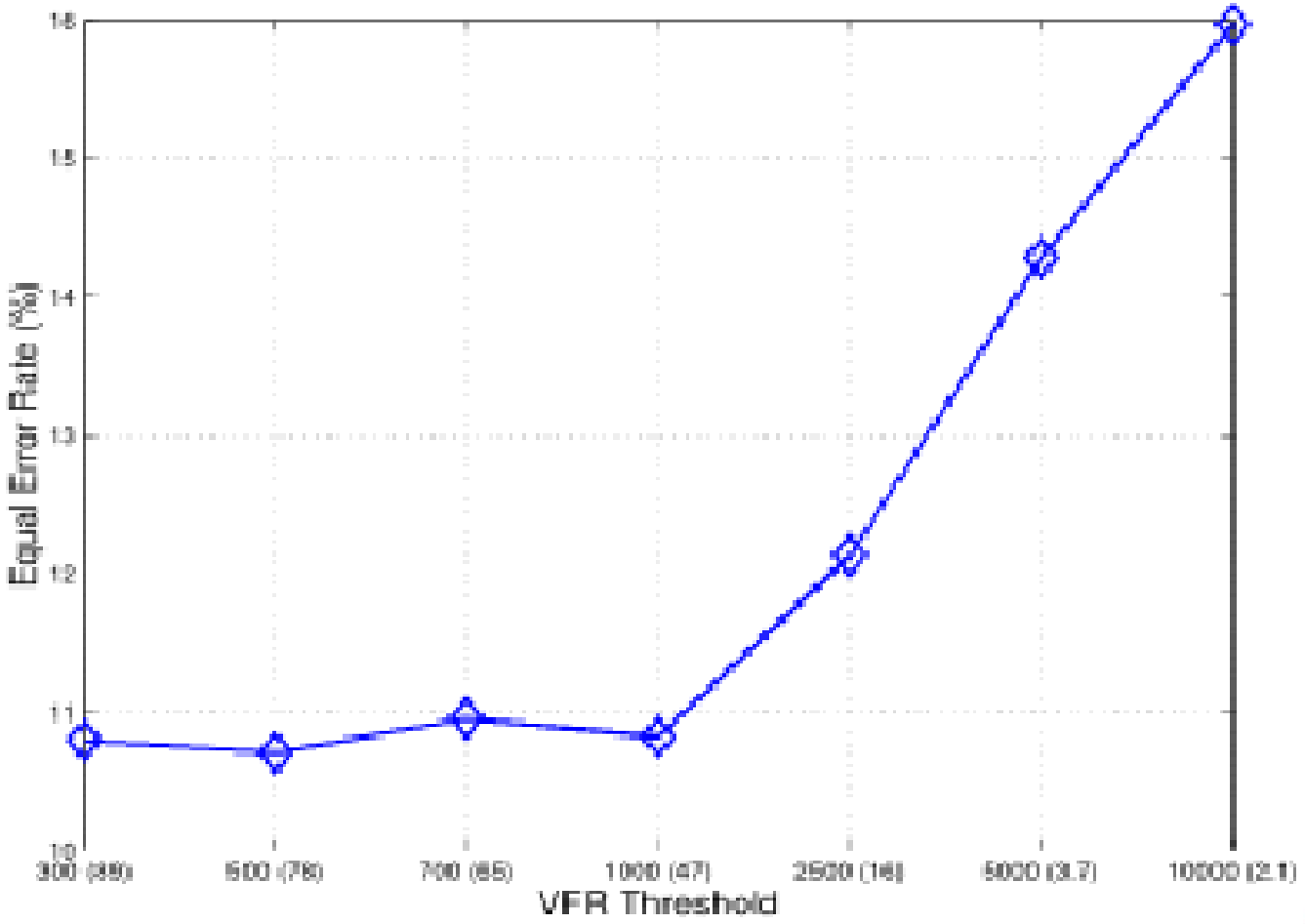
GMM-UBM, decymacja wektorów cech

- **Fixed-Rate Decimation vs EER (%)**



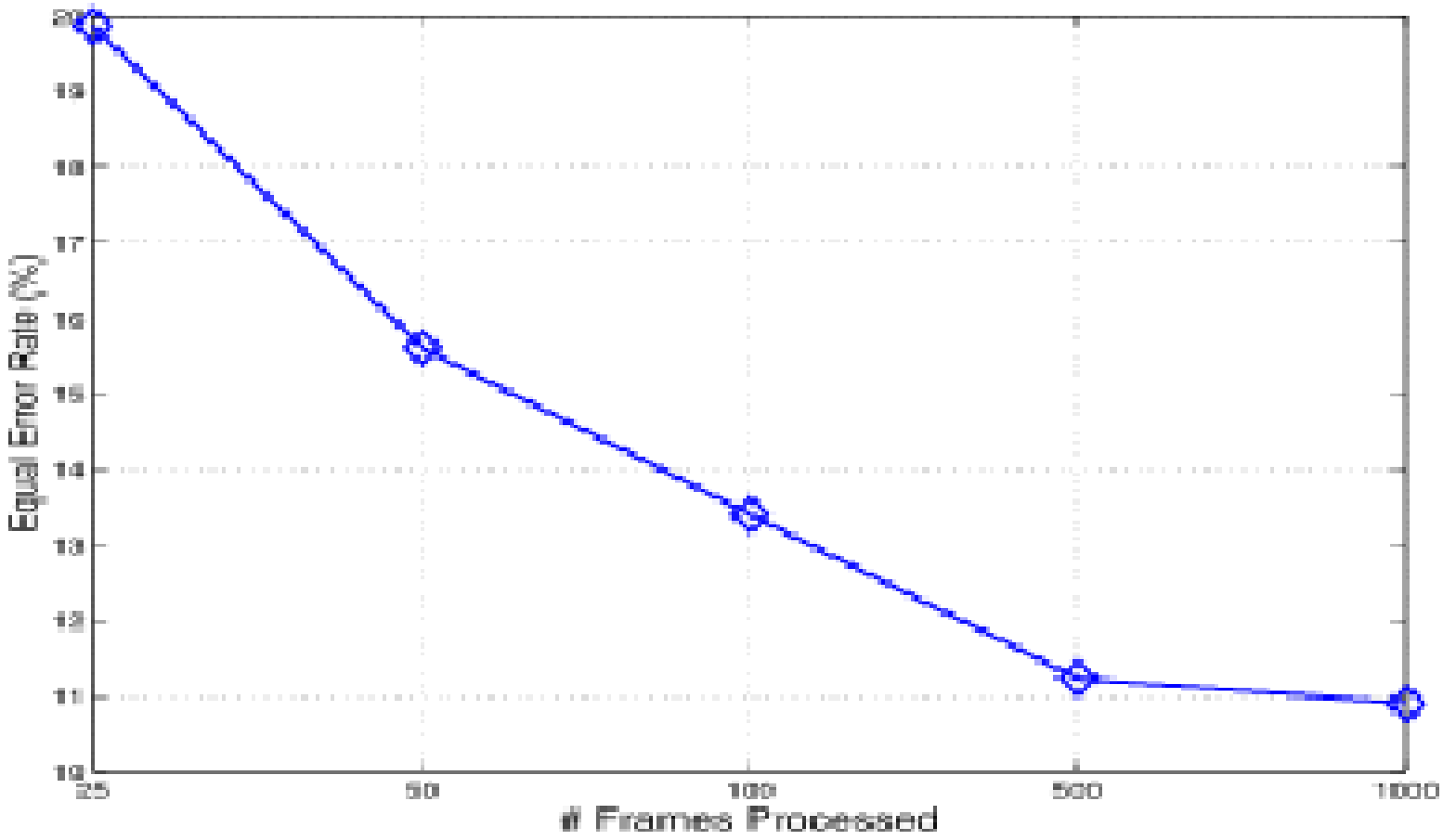
GMM-UBM, decymacja wektorów cech

- **Variable Frame Rate vs EER (%)**



GMM-UBM, decymacja wektorów cech

- **Adaptive-Rate Decimation vs EER (%)**



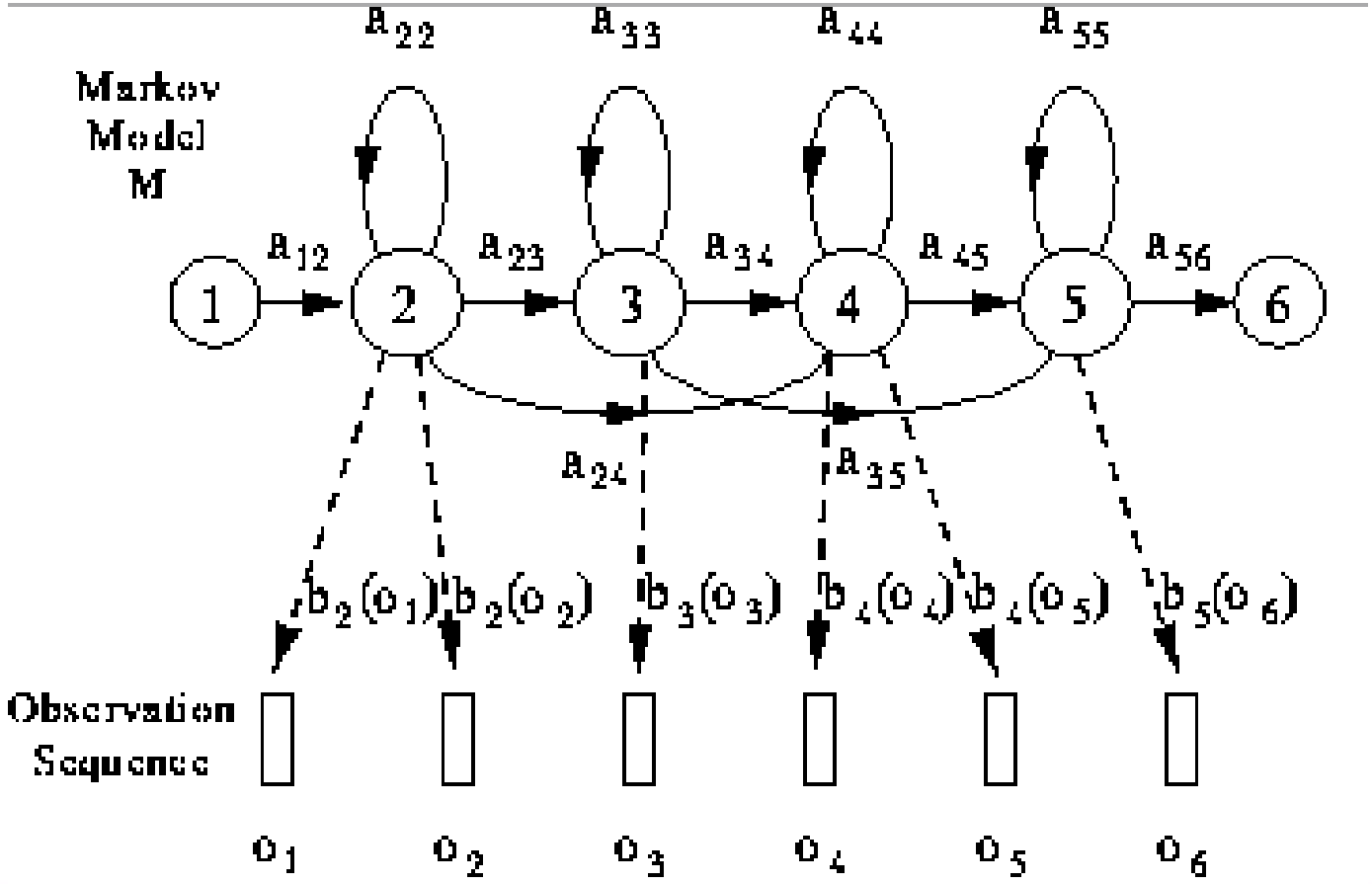
GMM-UBM, uwagi

- **Brak eksperymentów, które zbadałyby skuteczność zastosowania dwóch metod naraz (top-C + decymacja)**

- **Dotychczasowe testy przeprowadzono dla stosunkowo dużych modeli GMM (od 64 do 2048 mikstur) i długich wypowiedzi (od 3s do 15s) – aktualny system biometryczny wykorzystuje modele GMM zbudowane z 16 mikstur i długością hasła do 3s.**

HMM, krótki opis

- **System weryfikacji mówcy oparty o modelowanie HMM wymaga o wiele większej mocy obliczeniowej.**



HMM, krótki opis

- **Pojedynczy model HMM mówcy składa się z kilku/kilkunastu stanów emitujących, z których każdy posiada własną funkcję gęstości prawdopodobieństwa – GMM**

- **Do pełnego opisu modelu HMM mówcy wymagana jest macierz przejść i funkcje gęstości prawdopodobieństwa (GMM)**

- **Weryfikacja mówcy algorytmem Viterbiego lub algorytmem Forward**

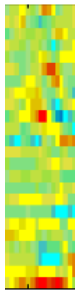
HMM, opis metody

- **Proces weryfikacji:**

$$\log(p(\vec{x}|\lambda_1) + \log(p(\vec{x}|\lambda_2) + \ldots + \log(p(\vec{x}|\lambda_N)$$

t = 1

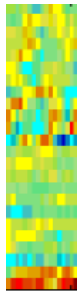
+



$$\log(p(\vec{x}|\lambda_1) + \log(p(\vec{x}|\lambda_2) + \ldots + \log(p(\vec{x}|\lambda_N)$$

t = 2

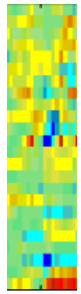
+



$$\log(p(\vec{x}|\lambda_1) + \log(p(\vec{x}|\lambda_2) + \ldots + \log(p(\vec{x}|\lambda_N)$$

t = 3

+ … +



$$\log(p(\vec{x}|\lambda_1) + \log(p(\vec{x}|\lambda_2) + \ldots + \log(p(\vec{x}|\lambda_N)$$

t = T

HMM, złożoność obliczeniowa

- **Główne czynniki wpływające na długość obliczeń:**
  - Ilość stanów emitujących
  - Ilość komponentów GMM dla każdego stanu
  - Wymiarowość wektora cech
  - Długość wypowiedzi

HMM, VQ-Based Gaussian Selection

- **Podobnie jak w przypadku GMM-UMB tak i w HMM techniki zwiększające wydajność obliczeniową systemu weryfikacji mówcy oparte są na zmniejszeniu ilości wykorzystywanych komponentów pochodzących z GMM**

- **Zazwyczaj dla jednego wektora cech tylko kilka, lub nawet jeden komponent dominuje rozkład prawdopodobieństwa GMM. Z tego wynika, że obliczenia mogą być ograniczone do jednego lub kilku komponentów i aproksymowane na pozostałe komponenty.**

- **Głównym celem metod optymalizacyjnych w przypadku HMM jest więc efektywne znalezienie tej grupy komponentów.**

HMM, VQ-Based Gaussian Selection

- **Metoda optymalizacji HMM oparta o kwantyzację wektorową (VQ) polega na podzieleniu wektorów cech na kilka podprzestrzeni (klasterów) i określeniu dla każdego klastra grupy komponentów maksymalizujących prawdopodobieństwo.**

- **Proces weryfikacji:**
  - Przydzielenie bieżącego wektora cech do jednego z wcześniej zdefiniowanych klastrów (metryka euklidesowa, itp.)

  - Wykorzystanie wcześniej przygotowanej grupy komponentów do obliczenia logarytmu prawdopodobieństwa

- **Poważny minus:** potrzebne duże zasoby pamięci

HMM, Partial Distance Elimination

- **Zamiast obliczać prawdopodobieństwo dla wszystkich komponentów, można w uproszczeniu przyjąć, że prawdopodobieństwo emisji danego stanu jest równe prawdopodobieństwu najlepiej rokującej mikstury. Z tej zasady korzysta metoda PDE.**

- **Proces weryfikacji:**
  - Należy obliczyć pełne prawdopodobieństwo dla pierwszej mikstury
  - Rozpocząć obliczenia dla kolejnej i przerwać je w momencie gdy wynik zacznie być gorszy od poprzedniego
  - Kontynuować obliczenia do momentu znalezienia najlepszego komponentu

HMM, Partial Distance Elimination

- **Zamiast obliczać prawdopodobieństwo dla wszystkich komponentów, można w uproszczeniu przyjąć, że prawdopodobieństwo emisji danego stanu jest równe prawdopodobieństwu najlepiej rokującej mikstury. Z tej zasady korzysta metoda PDE.**

- **Proces weryfikacji:**
  - Należy obliczyć pełne prawdopodobieństwo dla pierwszej mikstury
  - Rozpocząć obliczenia dla kolejnej i przerwać je w momencie gdy wynik zacznie być gorszy od poprzedniego
  - Kontynuować obliczenia do momentu znalezienia najlepszego komponentu

HMM, PDE + BMP + FER

- **Istnieją dwie techniki wspierające algorytm PDE:**

  1) Best Mixture Prediction – polega na zapamiętaniu najlepszej mikstury z poprzedniego wektora cech i rozpoczęciu od niej obliczeń na nowych wektorze

  2) Feature Element Reordering – zamiana kolejności elementów wektorów cech w ten sposób, by te, które silniej wpływają na rezultat obliczeń były wykorzystywane jako pierwsze

HMM, Dynamic Gaussian Selection

- **Łączy w sobie plusy VQ-BGS i PDE:**
  - Nie potrzebuje dodatkowej pamięci
  - Tworzy na bieżąco grupę najlepszych komponentów

- **Uproszczony schemat działania:**
  - Na początku oblicza wynik dla BMP
  - Następnie każdy komponent jest analizowany za pomocą algorytmu PDE
  - Po przekroczeniu określonej ilości iteracji algorytm przechodzi do następnego komponentu
  - Po uzyskaniu określonej ilości komponentów (zazwyczaj 5) algorytm kończy działanie

HMM

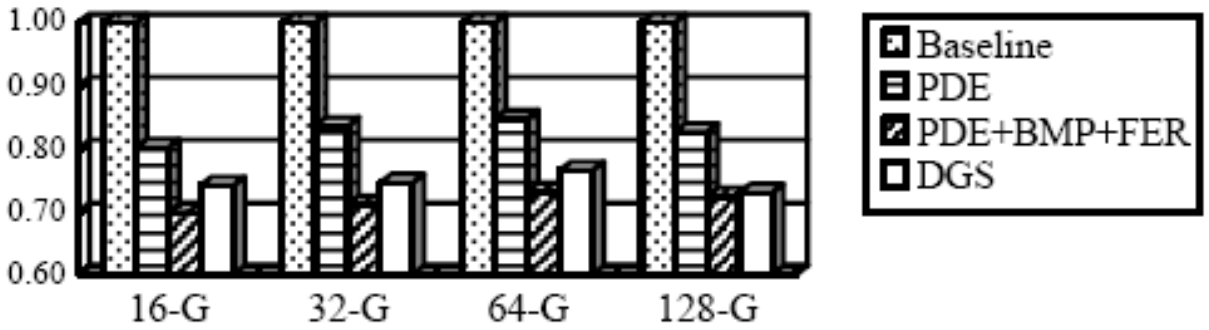


Figure 1 Normalized Recognition Time for TIMIT

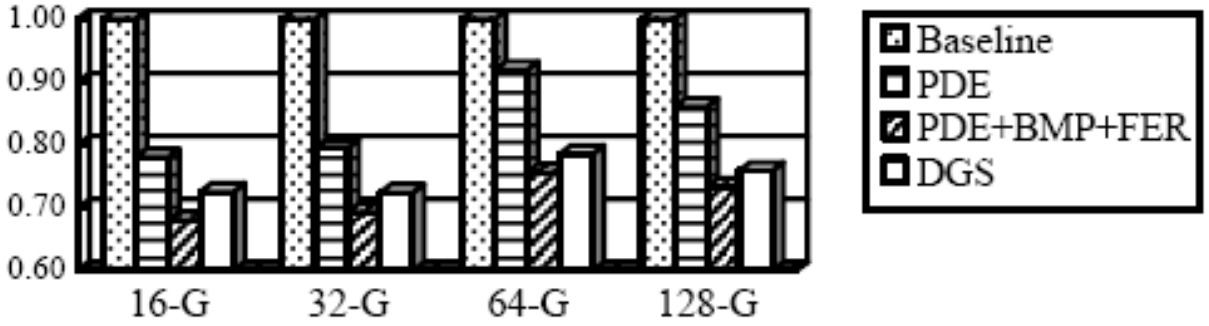


Figure 2 Normalized Recognition Time for HIWIRE

HMM

Table 1 Recognition Accuracy for TIMIT

Scheme	Phone Accuracy (%)			
	16-G	32-G	64-G	128-G
Baseline	56.7	58.7	59.9	60.2
PDE	56.3	58.3	59.4	59.7
PDE+BMP+FER	56.3	58.3	59.4	59.7
DGS	56.6	58.7	59.8	60.1

Table 2 Recognition Accuracy for HIWIRE

Scheme	Phone Accuracy (%)			
	16-G	32-G	64-G	128-G
Baseline	36.6	38.7	41.0	42.2
PDE	36.2	38.4	40.5	41.8
PDE+BMP+FER	36.2	38.4	40.5	41.8
DGS	36.6	38.8	41.0	42.2

Akceleracja sprzętowa i programowa

- **Rozwiązania zależne od platformy:**
  - NVIDIA CUDA – obliczenia równoległe
  - Cortex M4F – koprocesor zmiennoprzecinkowy
  - Itp..

- **Optymalizacja algorytmów podstawowych funkcji matematycznych:**
  - W szczególności exp, log
  - Z wykorzystaniem tablicowania (look-up table)

Dziękuję za uwagę!

Źródła:

[1] McLaughlin - A Study of Computation Speed-ups of the GMM-UBM Speaker Recognition System

[2] Kinnunen - Real-Time Speaker Identificatin and Verification

[3] Mohammadi - Efficient Implementation of GMM Based Speaker Verification Using Sorted GMM

[4] Cai - Dynamic Gaussian Selection Technique for Speeding Up HMM-Based Continuous Speech Recognition

[5] Bocchieri - Vector Quantization for the Efficient Computation of Continuous Density Likelihoods

[6] HTK BOOK