

ZASTOSOWANIE ALGORYTMÓW GENETYCZNYCH DO OPTIMALIZACJI MODELI HMM

STRESZCZENIE

W artykule przedstawiono metodę optymalizacji modeli HMM z wykorzystaniem algorytmu genetycznego. W celu zbadania skuteczności przedstawionego algorytmu genetycznego zostały przeprowadzone badania optymalizacji modeli HMM za pomocą algorytmu Bauma–Welcha oraz zaproponowanego algorytmu genetycznego. Dodatkowo w artykule zostały umieszczone wyniki z badań modelowania sygnału mowy w postaci przebiegów czasowych przez modele HMM optymalizowane algorytmem Bauma–Welcha.

Słowa kluczowe: optymalizacja, rozpoznawanie mowy, algorytmy genetyczne

THE USE OF GENETIC ALGORITHMS FOR OPTIMIZATION OF THE MODELS HMM

In the article there was presented the method of optimization of the models HMM with the use of the genetic algorithm. For the purpose of examining the effectiveness of the presented genetic algorithm there were carried out tests of optimization of the models HMM with the use of Baum–Welch's algorithm and the proposed genetic algorithm. In addition, in the article there were placed the results of tests of modeling the speech signal in the form of time runs by the models HMM optimized with the use of Baum–Welch's algorithm.

Keywords: optimization, speech recognition, genetic algorithms

1. WPROWADZENIE

W pracy przedstawiono metodę optymalizacji modeli HMM (hidden Markov model) z wykorzystaniem algorytmu genetycznego. W celu pokazania skuteczności przedstawionego algorytmu genetycznego zostały przeprowadzone badania modelowania mowy z zastosowaniem modeli HMM. Badania modelowania mowy przeprowadzono z zastosowaniem modeli HMM, optymalizowanych za pomocą standardowych metod (algorytm Bauma–Welcha) opisanych w rozdziale 3, oraz algorytmu genetycznego opisanego w rozdziale 4.

Badania modelowania mowy zostały przeprowadzone na podstawie sekwencji odpowiadających 15 sylabom wyodrębnionych z nagranych zdania. Sposób tworzenia sekwencji treningowych jest opisany w rozdziale 2. Do każdej z 15 sylab został dobrany model HMM. Metody doboru modeli HMM do sylab opisane są w rozdziałach 3 oraz 4. Następnie zostały policzone prawdopodobieństwa wygenerowania sekwencji treningowych innych niż sekwencje, do których były dobierane modele HMM. Porównanie wyników uzyskanych z optymalizacji modeli HMM z zastosowaniem metod standardowych oraz algorytmu genetycznego przedstawione są w rozdziale 5.

W modelu HMM określona jest liczba stanów m , stany numerujemy liczbami $\{1, 2, \dots, m\}$, dodatkowo stan w czasie t oznaczamy jako S_t . Liczba różnych obserwacji w stanie wynosi n , obserwacje numerujemy liczbami $\{1, 2, \dots, n\}$ oraz oznaczamy obserwacje w czasie t jako O_t . Parametry modelu HMM są opisane przez trójkę $M = (\Gamma, \pi, \delta)$. Macierz prawdopodobieństw przejść jest oznaczona przez $\Gamma = \{\gamma_{ij}\}$, gdzie γ_{ij} jest to prawdopodobieństwo przejścia ze

stanu i do stanu j . Macierz prawdopodobieństw wygenerowania kolorów oznaczona jest przez $\pi = \{\pi_{ki}\}$, gdzie π_{ki} jest to prawdopodobieństwo wygenerowania obserwacji k w węźle i . Wektor prawdopodobieństw początkowych oznaczony jest przez $\delta = \{\delta_i\}$, gdzie δ_i jest to prawdopodobieństwo początkowe w węźle i .

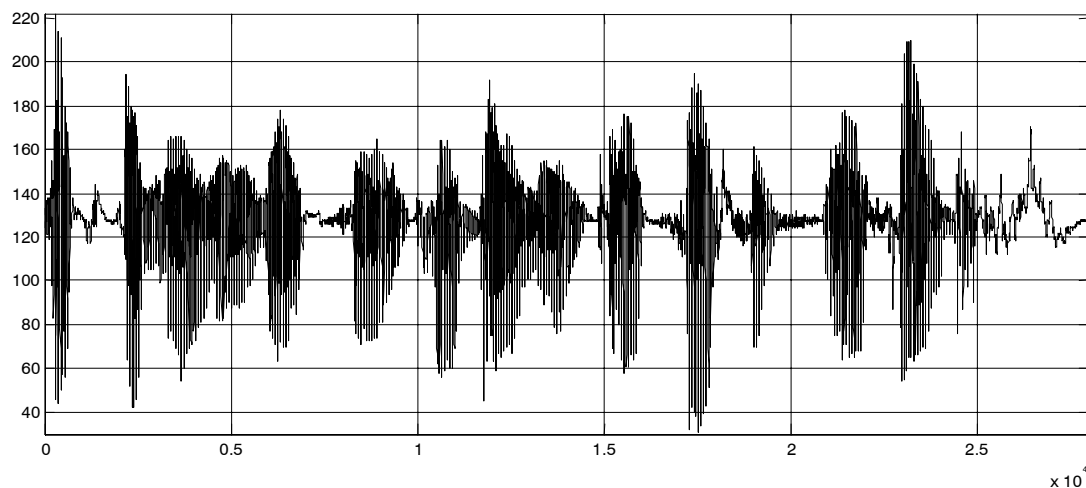
Sekwencje treningowe mowy reprezentowane są przez wektory obserwacji $\mathbf{O} = \{O_1, O_2, \dots, O_T\}$, obserwacje w chwili $t = 1, 2, \dots, T$ oznaczamy przez O_t .

2. TWORZENIE NOWYCH SEKWENCJI UCZĄCYCH

W systemach rozpoznawania mowy w celu uzyskania sekwencji, które następnie modeluje się za pomocą modeli HMM, dźwięk poddaje się odpowiedniej obróbce metodami takimi jak LPC (*Linear Predictive Coding*). W modelach HMM z dyskretnymi rozkładami prawdopodobieństw w stanach stosuje się dodatkowo metodę VC (*Vector Quantization*) pozwalającą na reprezentacje wektorów uzyskanych dzięki metodzie LPC w postaci pojedynczych wartości. Zadaniem stosowanych metod jest zmniejszenie liczby danych reprezentujących dźwięk oraz wyodrębnienie z dźwięku istotnych informacji.

Opisane w artykule badania modelowania zostały przeprowadzone bezpośrednio na czasowej postaci próbek nagranych dźwięku. Badania modelowania mowy zostały przeprowadzone na próbkach dźwięku nagranych za pomocą programu Cool Edit Pro. Nagrano sześć wersji zdania „Optymalizacja modeli procesów losowych”. Na rysunku 1 przedstawione jest jedno z nagrań.

* Studia doktoranckie, Wydział EAIiE AGH



Rys. 1. Próbką dźwięku nagrałego za pomocą programu Cool Edit Pro. Częstotliwość 8000 Hz, rozdzielczość 8 bitów na próbkę

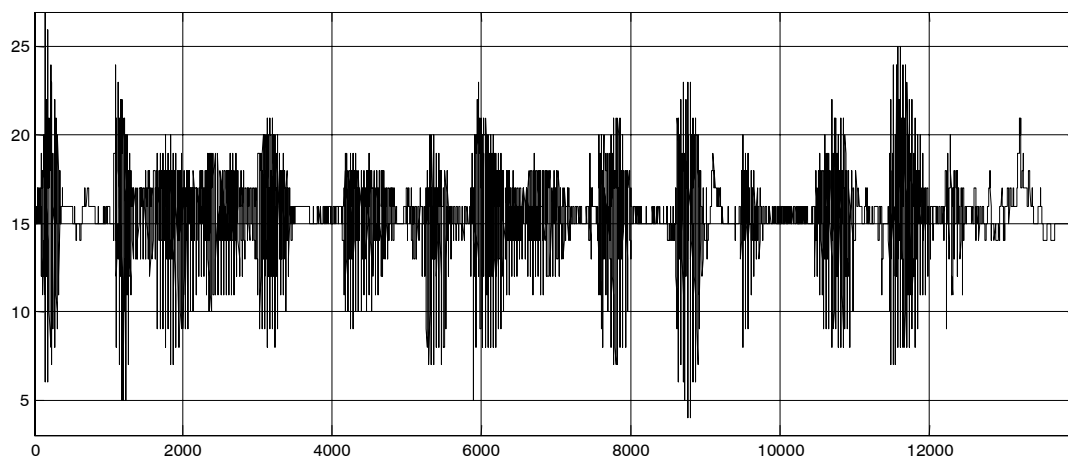
Dźwięk został nagrany w formacie (*wav*) z częstotliwością 8000 Hz i rozdzielczością 8 bitów na próbkę. Z nagranych zdań wyodrębniono sylaby, w wyniku tego otrzymano 45 sekwencji uczących po trzy sekwencje odpowiadające każdej sylabie. Częstotliwość próbkowania w sekwencjach uczących została zmieniona na 4000 Hz, natomiast rozdzielczość na 5 bitów na próbkę. Otrzymano sekwencje, w których liczba różnych obserwacji w stanie wynosi $n = 32$. Nowe wartości próbek przy zmianie częstotliwości zostały wyliczone poprzez linearyzację. Dodatkowo w sekwencjach uczących został podniesiony poziom dźwięku tak, aby największe wartości próbek w nagrany dźwięku miały maksymalne wartości w sekwencji uczącej. Rysunek 2 przedstawia dźwięk po wprowadzonych zmianach.

W dźwięku zostały dokonane zmiany, ponieważ w przypadku większej częstotliwości próbkowania oraz rozdzielczości prawdopodobieństwo uzyskania sekwencji odpowiadających sylabie są bardzo małe i przekraczają precyzje komputera. Przy takim formacie dźwięku zachowana jest

treść wypowiedzi, a wartości uzyskiwanych prawdopodobieństw są zadowalające.

Podczas badań stwierdzono, że sekwencje uczące zawierające dużo wartości na poziomie środkowym o wartości 16 (dla rozdzielczości 5 bitów) dają największe prawdopodobieństwa. Dzieje się tak, ponieważ w sekwencjach uczących reprezentujących dźwięk występuje wiele przejść przez wartość środkową. Po dobraniu modelu do jakiejś sekwencji treningowej największe prawdopodobieństwo uzyskujemy dla sekwencji zerowej, tj. takiej, w której wszystkie próbki mają wartości równe 16. Z powyższych powodów próbki dźwięku reprezentujące sylaby mają różne długości i nie mogą być rozszerzane przez wartości zerowe.

W celu porównania sekwencji o różnej długości nie można porównywać prawdopodobieństw wygenerowania tych sekwencji, dlatego zastosowana jest metoda opisana w rozdziale 3. Zależności (9) i (10) pozwalają obliczyć prawdopodobieństwo wygenerowania danej sekwencji na K próbek, niezależnie od długości danej sekwencji.



Rys. 2. Próbką dźwięku po obróbce. Częstotliwość próbkowania 4000 Hz, rozdzielczość 5 bitów na próbkę

3. METODY STANDARDOWE OPTIMALIZACJI MODELI HMM

W badaniach modelowania mowy zostały wykorzystane metody obliczania prawdopodobieństwa $P(\mathbf{O}|M)$ wygenerowania sekwencji $\mathbf{O} = \{O_1, O_2, \dots, O_T\}$ przez model $M = (\Gamma, \pi, \delta)$ jak w równaniach:

$$P(\mathbf{O} | M) = \sum_{i=1}^m \alpha_{\hat{i}}(i) \beta_{\hat{i}}(i), \quad \hat{i} = [2/T] \quad (1)$$

gdzie:

$$\alpha_{t+1}(j) = \left(\sum_{i=1}^m \alpha_t(i) \gamma_{ij} \right) \pi_{O_{t+1}j} \quad (2)$$

$$\beta_t(i) = \sum_{j=1}^m \pi_{O_{t+1}j} \beta_t(j) \gamma_{ij} \quad (3)$$

Zastosowano metodę Bauma–Welcha estymacji parametrów modelu HMM, jak w równaniach (4), (5) i (6). Podana metoda jest metodą skuteczniejszą niż metody gradientowe stosowane do optymalizacji modeli HMM, np.: w pracy [11], ponieważ w metodach gradientowych występują trudności ze znalezieniem odpowiedniej wartości kroku.

$$\hat{\delta}_i = \chi_1(i) = \frac{\alpha_1(i) \beta_1(i)}{\sum_{i=1}^m \alpha_i(i) \beta_i(i)} \quad (4)$$

$$\hat{\gamma}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \chi_t(i)} = \frac{\sum_{t=1}^{T-1} \alpha_t(i) \gamma_{ij} \pi_{O_{t+1}j} \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i) \beta_t(i)} \quad (5)$$

$$\hat{\pi}_{kj} = \frac{\sum_{t=1}^T \chi_t(j)}{\sum_{t=1}^T \chi_t(j)} = \frac{\sum_{t=1}^T \alpha_t(i) \beta_t(i)}{\sum_{t=1}^T \alpha_t(i) \beta_t(i)} \quad (6)$$

gdzie: $\chi_t(i)$, $\xi_t(i, j)$ opisane są zależnościami:

$$\begin{aligned} \chi_t(i) &= P(S_t = i | \mathbf{O}) = \frac{P(\mathbf{O} \wedge S_t = i)}{P(\mathbf{O})} = \\ &= \frac{P(\mathbf{O} \wedge S_t = i)}{\sum_{i=1}^m P(\mathbf{O} \wedge S_t = i)} = \frac{\alpha_t(i) \beta_t(i)}{\sum_{i=1}^m \alpha_t(i) \beta_t(i)} \end{aligned} \quad (7)$$

$$\begin{aligned} \xi_t(i, j) &= P(S_t = i, S_{t+1} = j | \mathbf{O}) = \\ &= \frac{P(S_t = i, S_{t+1} = j \wedge \mathbf{O})}{P(\mathbf{O})} = \frac{\alpha_t(i) \gamma_{ij} \pi_{O_{t+1}j} \beta_{t+1}(j)}{\sum_{i=1}^m \alpha_t(i) \beta_t(i)} \end{aligned} \quad (8)$$

Poza podanymi wyżej metodami w celu zmniejszenia błędów numerycznych zostały zastosowane metody ze skalowaniem opisane w podrozdziale 3.1.

W celu porównania prawdopodobieństw sekwencji treningowych o różnej długości nie można stosować metody wyznaczania prawdopodobieństwa według zależności (1). Stosowana jest zależność (9) pozwalająca wyznaczyć prawdopodobieństwo na próbkę oraz zależność (10) pozwalająca wyznaczyć prawdopodobieństwo dla K próbek:

$$P_{\text{próbki}}(\mathbf{O} | M) = P(\mathbf{O} | M)^{1/T} \quad (9)$$

$$P_K(\mathbf{O} | M) = P(\mathbf{O} | M)^{K/T} \quad (10)$$

3.1. Skalowanie

Aby zrozumieć, dlaczego stosujemy skalowanie w procedurze estymacji modelu HMM, przyjrzyjmy się definicji współczynników $\alpha_t(i)$. Można zauważyć, że $\alpha_t(i)$ składa się z sumy dużej liczby wyrażeń, gdzie każde ma postać jak w zależności

$$\left(\prod_{k=1}^{t-1} \gamma_{S_k S_{k+1}} \prod_{k=1}^t \pi_{O_k S_k} \right) \quad (11)$$

Wartości γ_{ij} oraz π_{O_i} są znacznie mniejsze od jedynki, stąd dla t większych od 10 wartość $\alpha_t(i)$ zmierza eksponentalnie do zera. Dla t większych od 100 wartość $\alpha_t(i)$ może przekroczyć precyzję komputera. Stąd jedynym sposobem poradzenia sobie z tym problemem jest zastosowanie procedury ze skalowaniem.

Prosta procedura skalowania powiela $\alpha_t(i)$ o czynnik, który jest niezależny od stanu i (jest zależny jedynie od czasu t) w celu utrzymania skalowanej wartości $\alpha_t(i)$ w pewnym przedziale dla czasu $1 \leq t \leq T$. Podobna procedura jest zastosowana do skalowania $\beta_t(i)$ (wartości $\beta_t(i)$ również dążą do zera eksponentalnie). Po zastosowaniu metody powielania o czynnik, na końcu obliczeń czynnik, o który powielamy, redukuje się.

Aby lepiej zrozumieć tę procedurę, rozważmy formułę estymacji na współczynnik przejścia $\hat{\gamma}_{ij}$. Jeżeli zapiszemy tę formułę w zależności od $\alpha_t(i)$ oraz $\beta_t(i)$, dostaniemy wyrażenie opisane równaniem

$$\hat{\gamma}_{ij} = \frac{\sum_{t=1}^{T-1} \alpha_t(i) \gamma_{ij} \pi_{O_{t+1}j} \beta_{t+1}(i)}{\sum_{t=1}^{T-1} \sum_{j=1}^n \alpha_t(i) \gamma_{ij} \pi_{O_{t+1}j} \beta_{t+1}(i)} \quad (12)$$

Rozważmy wyliczanie $\alpha_t(i)$. Użyjemy notacji $\alpha_t(i)$ w celu oznaczenia współczynnika bez skalowania oraz $\hat{\alpha}_t(i)$ ze skalowaniem. Dodatkowo oznaczymy przez $\hat{\hat{\alpha}}_t(i)$ lokalny współczynnik przed skalowaniem. Początkowo wyliczymy dla $t = 1$ $\alpha_t(i)$, według równania podstawowego, oraz przyjmujemy, że $\hat{\hat{\alpha}}_1(i) = \alpha_1(i)$, oraz wyznaczamy współczynnik $\hat{\alpha}_1(i)$ po skalowaniu według zależności $\hat{\alpha}_1(i) = c_1 \alpha_1(i)$, gdzie $c_1 = \frac{1}{\sum_{i=1}^N \alpha_1(i)}$.

Dla każdego t , $2 \leq t \leq T$ najpierw wyliczymy $\hat{\hat{\alpha}}_t(i)$ według równania (2), w zależności od poprzednio skalowanych wartości $\hat{\alpha}_t(i)$ jak w równaniu

$$\hat{\hat{\alpha}}_t(i) = \sum_{j=1}^n \hat{\alpha}_{t-1}(j) \gamma_{ji} \pi_{O_i} \quad (13)$$

Współczynnik skalowania c_t wyliczany jest według równania

$$c_t = \frac{1}{\sum_{i=1}^N \hat{\alpha}_t(i)} \quad (14)$$

Skalowane $\hat{\alpha}_t(i)$ wyznaczamy z równania

$$\hat{\alpha}_t(i) = c_t \hat{\hat{\alpha}}_t(i) \quad (15)$$

Korzystając z równań (13) i (14) $\hat{\alpha}_t(i)$, możemy zapisać

$$\hat{\alpha}_t(i) = \frac{\sum_{j=1}^N \hat{\alpha}_{t-1}(j) \gamma_{ji} \pi_{O_i}}{\sum_{i=1}^N \sum_{j=1}^N \hat{\alpha}_{t-1}(j) \gamma_{ji} \pi_{O_i}} \quad (16)$$

Można wykazać indukcyjnie, że zachodzi równość

$$\hat{\alpha}_{t-1}(j) = \left(\prod_{\tau=1}^{t-1} c_\tau \right) \alpha_{t-1}(j) \quad (17)$$

W ten sposób możemy zapisać $\hat{\alpha}_t(i)$ jak w równaniu

$$\hat{\alpha}_t(j) = \frac{\sum_{j=1}^N \alpha_{t-1}(j) \left(\prod_{\tau=1}^{t-1} c_\tau \right) \gamma_{ji} \pi_{O_i}}{\sum_{i=1}^N \sum_{j=1}^N \alpha_{t-1}(j) \left(\prod_{\tau=1}^{t-1} c_\tau \right) \gamma_{ji} \pi_{O_i}} = \frac{\alpha_t(i)}{\sum_{i=1}^N \alpha_t(i)} \quad (18)$$

stąd wynika, że $\alpha_t(i)$ jest efektywnie skalowane przez sumę.

Podczas wyliczania $\beta_t(i)$ używa się tych samych współczynników skalowania, które były wyznaczone przy wyliczaniu $\alpha_t(i)$, stąd skalowanie przybiera formę jak w równaniu

$$\hat{\beta}_t(i) = c_t \hat{\hat{\beta}}_t(i) \quad (19)$$

Ponieważ wektor skalowania c_t pozostaje po skalowaniu α , a rozmiary α oraz β są porównywalne. Stąd użycie tego samego wektora jest efektywne w celu uzyskania rozsądnych wartości. Po zastosowaniu skalowania wartości współczynników macierzy przejść wyznaczamy z równania

$$\hat{\gamma}_{ij} = \frac{\sum_{t=1}^{T-1} \hat{\alpha}_t(i) \gamma_{ij} \pi_{O_{t+1}j} \hat{\beta}_{t+1}(i)}{\sum_{t=1}^{T-1} \sum_{j=1}^N \hat{\alpha}_t(i) \gamma_{ij} \pi_{O_{t+1}j} \hat{\beta}_{t+1}(i)} \quad (20)$$

Każde $\alpha_t(i)$ może być zapisane jak w równaniu

$$\hat{\alpha}_t(i) = \left(\prod_{k=1}^t c_k \right) \alpha_1(i) = C_t \alpha_t(i) \quad (21)$$

Natomiast $\beta_t(i)$ może być zapisane jak w równaniu

$$\hat{\beta}_{t+1}(j) = \left(\prod_{k=t+1}^T c_k \right) \beta_{t+1}(j) = D_{t+1} \beta_{t+1}(j) \quad (22)$$

Tak więc estymator (12) możemy wyrazić równaniem

$$\hat{\gamma}_{ij} = \frac{\sum_{t=1}^{T-1} C_t \alpha_t(i) \gamma_{ij} \pi_{s_{t+1}j} D_{t+1} \beta_{t+1}(i)}{\sum_{t=1}^{T-1} \sum_{j=1}^N C_t \alpha_t(i) \gamma_{ij} \pi_{s_{t+1}j} D_{t+1} \beta_{t+1}(i)} \quad (23)$$

W efekcie $C_t D_{t+1}$ możemy zapisać jako C_T jak w równaniu

$$C_t D_{t+1} = \prod_{k=1}^t c_k \prod_{k=t+1}^T c_k = \prod_{k=1}^T c_k = C_T \quad (24)$$

Wartość C_T jest niezależna od t . W równaniu (23) wartości $C_t D_{t+1}$ skracają się, stąd wynik jest realizacją estymacji.

Powyższe skalowanie zastosowane do estymacji macierzy prawdopodobieństw kolorów π oraz wektora stanów początkowych δ daje podobne rezultaty. Również powinno być oczywiste, że procedura skalowania nie powinna być stosowana w każdym czasie t , ale powinna być wykonywana, gdy jest konieczna. Jeżeli skalowanie nie jest wykonywane dla pewnych czasów t , wówczas c_t przyjmuje wartość 1, oraz wszystkie powyższe zależności są spełnione.

Jedyną realną zmianą w modelu HMM spowodowaną skalowaniem jest procedura wyznaczania $P(\mathbf{O}|\mathbf{M})$. Możemy jedynie zsumować $\hat{\alpha}_T(i)$, ponieważ te były wcześniej skalowane. W wyniku sumowania $\hat{\alpha}_T(i)$ dostajemy równanie

$$\prod_{t=1}^T c_t \sum_{i=1}^N \alpha_T(i) = C_T \sum_{i=1}^N \alpha_T(i) = 1 \quad (25)$$

Równanie (25) możemy zapisać w postaci pokazanej w równaniach:

$$\prod_{t=1}^T c_t P(\mathbf{O}|\mathbf{M}) = 1$$

$$P(\mathbf{O}|\mathbf{M}) = \frac{1}{\prod_{t=1}^T c_t} \quad (26)$$

$$\log[P(\mathbf{O}|\mathbf{M})] = -\sum_{t=1}^T \log c_t$$

Podczas stosowania procedury skalowania nie może być obliczone P , ale $\log P$. Aby można było zastosować zależności (26), należy skalować $\alpha_T(i)$.

W efekcie dostajemy równania na estymator wyliczane ze skalowanych współczynników $\hat{\alpha}_t(i)\hat{\beta}_t(i)$, $t = 1, 2, \dots, T$, $i = 1, 2, \dots, m$, jak poniżej:

$$\hat{\delta}_i = \frac{\hat{\alpha}_1(i)\hat{\beta}_1(i)}{\sum_{i=1}^m \hat{\alpha}_1(i)\hat{\beta}_1(i)} \quad (27)$$

$$\hat{\gamma}_{ij} = \frac{\sum_{t=1}^{T-1} \hat{\alpha}_t(i)\gamma_{ij}\pi_{s_{t+1}j}\hat{\beta}_{t+1}(j)}{\sum_{t=1}^{T-1} \hat{\alpha}_t(i)\hat{\beta}_t(i)/c_t} \quad (28)$$

$$\hat{\pi}_{kj} = \frac{\sum_{t=1}^T \hat{\alpha}_t(j)\hat{\beta}_t(j)}{\sum_{t=1}^T \hat{\alpha}_t(j)\hat{\beta}_t(j)} \quad (29)$$

3.2. Algorytmy optymalizacji

Aby odpowiednio dobrać parametry modelu HMM, by reprezentował sylabę, musimy optymalizować go do zbioru sekwencji. Zastosowano trzy metody doboru parametrów

do zbioru sekwencji, w którym każda sekwencja odpowiada tej samej sylabie. Zbiór sekwencji oznaczamy przez

$$\mathbf{Z} = \{\mathbf{O}_1, \mathbf{O}_2, \dots, \mathbf{O}_z\},$$

gdzie:

z – liczba sekwencji w zbiorze,

$\mathbf{O}_k = (O_1^{(k)}, O_2^{(k)}, \dots, O_{T_k}^{(k)})$ – k -ta sekwencja.

Metoda pierwsza podana w literaturze [1] stosowana jest najczęściej w modelach HMM nazywanych z „lewa na prawo”, w których występuje duża liczba węzłów oraz pewne prawdopodobieństwa przejść są zerowe, w modelach takich przy optymalizacji każdemu z węzłów odpowiada niewielka liczba obserwacji. Żeby dobrać odpowiednio parametry modelu, musimy mieć wystarczającą liczbę danych. Stosuje się więc wielokrotne sekwencje obserwacji odpowiadające danemu słowu. Metody estymacji do wielokrotnych sekwencji mogą być również z powodzeniem stosowane do zwykłych modeli HMM.

Naszym zadaniem jest minimalizacja wyrażenia

$$P(\mathbf{Z}|\mathbf{M}) = \prod_{k=1}^Z P(\mathbf{O}_k|\mathbf{M}) = \prod_{k=1}^Z P_k \quad (30)$$

Wówczas równania na estymatory parametrów przyjmują postać jak w równaniach:

$$\hat{\delta}_i = \frac{\sum_{k=1}^z \frac{1}{P_k} \alpha_1^k(i)\beta_1^k(i)}{\sum_{k=1}^z \frac{1}{P_k} \sum_{i=1}^m \alpha_1^k(i)\beta_1^k(i)} \quad (31)$$

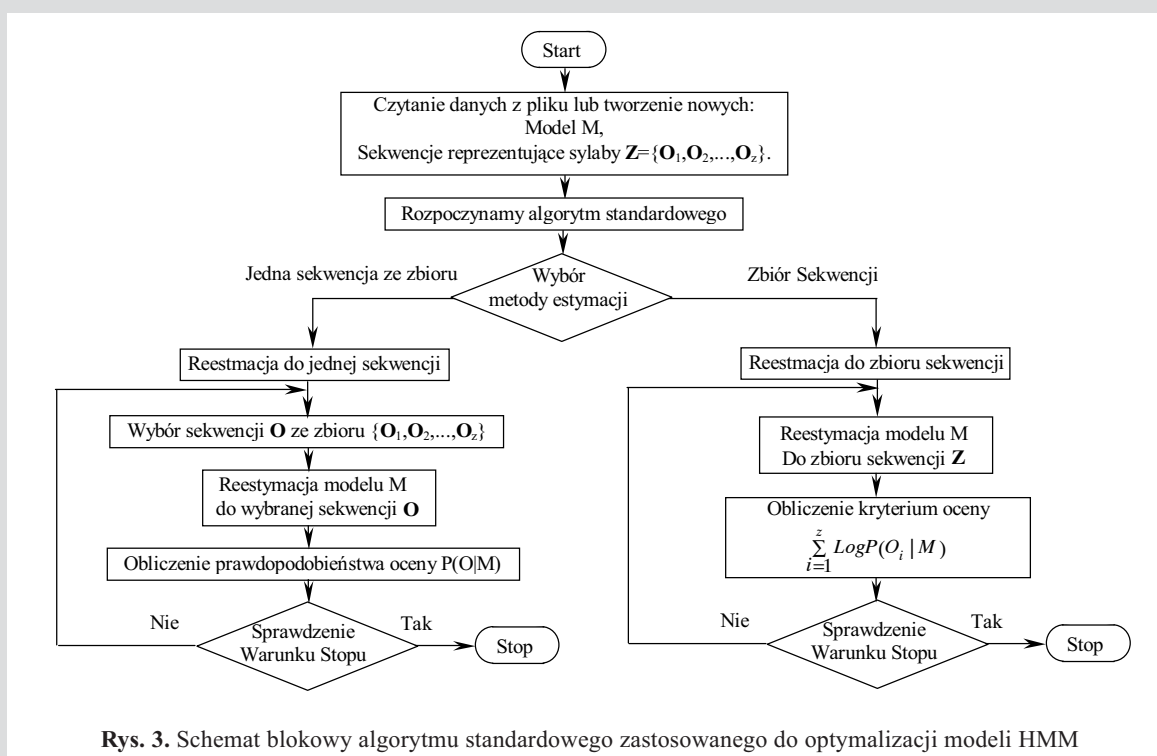
$$\hat{\gamma}_{ij} = \frac{\sum_{k=1}^z \frac{1}{P_k} \sum_{t=1}^{T-1} \alpha_t^k(i)\gamma_{ij}\pi_{O_{t+1}j}\beta_{t+1}^k(j)}{\sum_{k=1}^z \frac{1}{P_k} \sum_{t=1}^{T-1} \alpha_t^k(i)\beta_t^k(i)} \quad (32)$$

$$\hat{\pi}_{li} = \frac{\sum_{k=1}^z \frac{1}{P_k} \sum_{(po \ t \ t.z. O_t=l)} \alpha_t^k(i)\beta_t^k(i)}{\sum_{k=1}^z \frac{1}{P_k} \sum_{t=1}^T \alpha_t^k(i)\beta_t^k(i)} \quad (33)$$

W drugiej metodzie w każdym kroku algorytmu optymalizacji losowana jest jedna sekwencja ze zbioru sekwencji, następnie dobór parametrów przeprowadzany jest do wylosowanej sekwencji. Wybór sekwencji przeprowadzany jest jak w zależnościach

$$\mathbf{O} = \mathbf{O}_k \quad (34)$$

gdzie $k = \text{random}\{1, 2, \dots, z\}$.



Rys. 3. Schemat blokowy algorytmu standardowego zastosowanego do optymalizacji modeli HMM

W trzeciej metodzie w każdej iteracji wybierana jest sekwencja dająca najmniejsze prawdopodobieństwo zaobserwowania tej sekwencji, następnie dobór parametrów modelu przeprowadzany jest do wybranej sekwencji. Wybór sekwencji przeprowadzany jest jak w zależnościach

$$\mathbf{O} = \mathbf{O}_k \quad (35)$$

gdzie $k = \arg \min_{i=1, \dots, z} \{P(\mathbf{O}_i | M)\}$.

Na rysunku 3 przedstawiony jest schemat blokowy zastosowanego algorytmu do optymalizacji modeli HMM.

Jako warunek stopu przyjęto liczbę wykonanych iteracji, liczbę bez zmiany prawdopodobieństwa dla wybranej sekwencji, czas obliczeń oraz uzyskanie prawdopodobieństwa równego jeden dla wybranej sekwencji.

W algorytmie standardowym zadbano również o to, żeby wagi modelu nie przyjmowały wartości równych 0, co może się zdarzyć w procedurze estymacji z powodu przekroczenia precyzji komputera przez wyliczane parametry i dlatego przyjęto pewne graniczne wartości minimalne dla parametrów δ_i , γ_{ij} , π_{si} modelu. Wielkości graniczne parametrów zawierają się w przedziale $10^{-4} \div 10^{-100}$. W procedurze estymacji, jeżeli wartość jakiegoś parametru modelu spadnie poniżej ustalonego progu, wówczas wartościom tych parametrów przypisywana jest wartość progowa, następnie, jeśli taka sytuacja miała miejsce, przeprowadzana jest procedura normalizacji jak w równaniach (36)–(38).

4. ALGORYTM GENETYCZNY

W tym rozdziale przedstawiony zostanie algorytm genetyczny optymalizacji modeli HMM, dobierający model do

zbioru sekwencji odpowiadającej danej sylabie. W wyniku zastosowania algorytmu Bauma–Welcha do optymalizacji modeli HMM uzyskiwane są optima lokalne. Celem zastosowania algorytmu genetycznego ma być wychodzenie z uzyskanych optimów lokalnych.

W przedstawionym algorytmie, tak jak w większości algorytmów genetycznych, wykorzystane zostały podstawowe procedury algorytmów genetycznych, tj: reprodukcja, krzyżowanie oraz mutacja. Zaproponowany algorytm genetyczny działa krokowo. W pierwszym kroku tworzona jest populacja początkowa, w której każdy osobnik jest reprezentowany przez model HMM zakodowany w postaci chromosomu, w każdym następnym kroku algorytmu w procedurze reprodukcji z krzyżowaniem tworzone są osobniki potomne oraz poddawane procedurze mutacji. Z populacji początkowej oraz osobników potomnych w procedurze selekcji tworzona jest nowa populacja. W procedurach reprodukcji z krzyżowaniem oraz selekcji dokonywany jest wybór osobników ze względu na wartość ich funkcji przystosowania.

W dalszej części tego rozdziału opisane jest tworzenie nowej populacji, kodowanie chromosomu, funkcja oceny, procedury reprodukcji i krzyżowania oraz procedura mutacji.

4.1. Tworzenie nowej populacji

Nowe rozwiązania w populacji tworzone są w sposób losowy bądź niektóre rozwiązania mogą być dodawane do populacji po wcześniejszej optymalizacji za pomocą metod standardowych opisanych w rozdziale 3. W algorytmie ustalona jest liczba rozwiązań w populacji, oznaczamy ją przez x .

Rozwiązania losowe są tworzone w następujący sposób. W programie ustalona jest pewna liczba całkowita L_{max} , na podstawie przyjętej wartości losowane są liczby całkowite z przedziału $< 0, L_{max} >$ i wszystkim parametrom modelu przypisywane są wartości losowe z tego przedziału. Następnie HMM modyfikowany jest w sposób zapewniający spełnienie ograniczeń na parametry. Modyfikacja przeprowadzana jest jak w równaniach:

$$\hat{\delta}_i = \frac{\bar{\delta}_i}{\sum_{i=1}^m \bar{\delta}_i} \quad (36)$$

$$\hat{\gamma}_{ij} = \frac{\bar{\gamma}_{ij}}{\sum_{i=1}^m \bar{\gamma}_{ij}}, \quad j = 1, 2, \dots, m \quad (37)$$

$$\hat{\pi}_{S_{kj}} = \frac{\bar{\pi}_{S_{kj}}}{\sum_{O_k \in \{1, 2, \dots, n\}} \bar{\pi}_{S_{kj}}}, \quad j = 1, 2, \dots, m \quad (38)$$

W wyniku powyższej normalizacji otrzymujemy losowy model HMM, w którym wagi modelu mogą uzyskać wartość prawdopodobieństwa z przedziału od 0 do 1.

4.2. Kodowanie chromosomów

Kodowanie chromosomów osobnika M przeprowadzone jest w sposób pokazany w poniższej zależności

Chromosom:

$$M = (\delta_i \ \gamma_{1j} \ | \ \gamma_{2j} \ \gamma_{mj} \ \pi_{k1} \ | \ \pi_{k2} \ \pi_{km}) \quad (39)$$

$$i, j = 1, 2, \dots, m \quad k = 1, 2, \dots, n$$

Chromosom reprezentowany jest przez wektor prawdopodobieństw początkowych $\{\delta_i\}$, wektory prawdopodobieństw przejść z węzła pierwszego $\{\gamma_{1j}\}$, węzła drugiego $\{\gamma_{2j}\}$ aż do węzła m -tego $\{\gamma_{mj}\}$ oraz wektory prawdopodobieństw wylosowania kolorów w węzle pierwszym $\{\pi_{k1}\}$, węzle drugim $\{\pi_{k2}\}$ aż do węzła m -tego $\{\pi_{km}\}$, gdzie $i, j = 1, 2, \dots, m, k = 1, 2, \dots, n$.

4.3. Funkcja oceny

W każdym kroku algorytmu genetycznego stosowana jest ocena osobników populacji oraz osobników potomków. Wartość funkcji oceny oznaczamy przez $Q(M)$. Zastosowane zostały trzy metody oceny.

W pierwszej metodzie wybierana jest losowo sekwencja \mathbf{O} reprezentująca sylabę, następnie liczona jest wartość funkcji oceny jako prawdopodobieństwo wygenerowania wybranej sekwencji jak w równaniu

$$Q(M) = P(\mathbf{O} | M) \quad (40)$$

W drugiej metodzie wartość funkcji oceny przyjmuje minimalną wartość prawdopodobieństwa jak w zależności

$$Q(M) = \min P(O_i | M), \quad i = 1, 2, \dots, z \quad (41)$$

W trzeciej metodzie oceny wartość funkcji dla osobnika jest to iloczyn logarytmów prawdopodobieństw wygenerowania poszczególnych sekwencji ze zbioru sekwencji jak w zależności

$$Q(M) = \sum_{i=1}^z \log P(O_i | M) \quad (42)$$

4.4. Reprodukacja z krzyżowaniem

Modele HMM reprezentujące rodziców wybierane są w sposób częściowo losowy, ale z uwzględnieniem wartości funkcji oceny. Z populacji wybierane są dwa rozwiązania w sposób losowy. Pierwsze o lepszej wartości funkcji przystosowania przyjmowane jest jako pierwszy rodzic. Proces ten jest powtarzany w celu wyodrębnienia drugiego rodzica.

Rozwiązania potomków są tworzone z wykorzystaniem rozwiązań rodziców według standardowej procedury krzyżowania, w wyniku której dostajemy rozwiązania potomne. W algorytmie została zastosowana dwupunktowa operacja krzyżowania, w której dwa punkty krzyżowania wybierane są w sposób losowy. Pierwszy potomek tworzony jest z genów rodzica pierwszego na prawo od pierwszego punktu krzyżowania oraz na lewo od drugiego punktu krzyżowania oraz z genów drugiego z rodziców, które są pomiędzy dwoma punktami krzyżowania. Drugi potomek tworzony jest w ten sam sposób, tylko rodzice brani są w odwrotnej kolejności. W zależnościach (43) i (44) przedstawiony jest przykład operacji krzyżowania na przykładzie dwóch rodziców M oraz \tilde{M} :

$$M = (\delta_i \ \gamma_{1j} \ | \ \gamma_{2j} \ \gamma_{mj} \ \pi_{k1} \ | \ \pi_{k2} \ \pi_{km}) \quad (43)$$

$$\tilde{M} = (\tilde{\delta}_i \ \tilde{\gamma}_{1j} \ | \ \tilde{\gamma}_{2j} \ \tilde{\gamma}_{mj} \ \tilde{\pi}_{k1} \ | \ \tilde{\pi}_{k2} \ \tilde{\pi}_{km})$$

Punkty krzyżowania zostają wyznaczone w sposób losowy, na przykładzie są one zaznaczone pionowymi liniami. Pierwszy punkt krzyżowania umieszczony jest pomiędzy wektorem prawdopodobieństw przejść z pierwszego węzła a wektorem prawdopodobieństw przejść z drugiego węzła, drugi – pomiędzy wektorem wylosowania kolorów w pierwszym węzle a wektorem wylosowania kolorów w drugim węzle.

W wyniku krzyżowani otrzymujemy dwa osobniki potomków M' oraz \tilde{M}' :

$$\begin{aligned} M' &= (\delta_i \ \gamma_{1j} \mid \tilde{\gamma}_{2j} \ \tilde{\gamma}_{mj} \ \tilde{\pi}_{k1} \mid \pi_{k2} \ \pi_{km}) \\ \tilde{M}' &= (\tilde{\delta}_i \ \tilde{\gamma}_{1j} \mid \gamma_{2j} \ \gamma_{mj} \ \pi_{k1} \mid \tilde{\pi}_{k2} \ \tilde{\pi}_{km}) \end{aligned} \quad (44)$$

Należy zaznaczyć że w wyniku operacji krzyżowania otrzymujemy parzystą liczbę potomków.

4.5. Mutacja

Operacja mutacji przeprowadzana jest w każdym kroku algorytmu. Mutacji poddawane są rozwiązania potomków. W algorytmie przyjęto parametr mutacji z zakresu $0,001 \div 0,5$ określający prawdopodobieństwo, z jakim poddawana jest modyfikacji każda składowa genu odpowiadająca wadze modelu HMM. Według przyjętego prawdopodobieństwa parametry modeli, tj. prawdopodobieństwa początkowe δ_i , przejść γ_{ij} , oraz kolorów π_{ki} poddawane są modyfikacji. Losowana jest liczba z przedziału od 0 do L_{max} , następnie parametr poddawany modyfikacji przyjmuje wylosowaną wartość. Jeżeli nastąpiła zmian parametrów modelu, wówczas stosowana jest normalizacji jak w zależnościach (36)–(38).

4.6. Selekcja

W procedurze selekcji osobniki nowej populacji mogą pochodzić zarówno ze zbioru potomków, jak i populacji początkowej lub tylko ze zbioru potomków. Grupa osobników, z której tworzona jest nowa populacja, nazwiemy grupą bazową.

W selekcji wykorzystywane są dwie metody wyboru osobników do nowej populacji z grupy bazowej:

- 1) z grupy bazowej wybierane są najlepsze osobniki według zastosowanej wartości funkcji oceny,
- 2) stosowana jest metoda grupowania osobników według wartości funkcji przystosowania.

Dzięki zastosowaniu pierwszej metody zapewnione są dobre wartości rozwiązań, ale może ona powodować lokalną zbieżność algorytmu. W drugiej metodzie wyboru stosowane jest grupowanie osobników. Osobniki z grupy bazowej dzielone są na trzy grupy, przynależność do każdej z grup zależy od wartości funkcji przystosowania, np. najlepiej przystosowane 40%, mniej przystosowane 40% oraz najgorzej przystosowane 20%. Po grupowaniu osobniki do nowej populacji są wybierane z każdej grupy w odpowiednich z góry określonych proporcjach, np.: 40% z pierwszej grupy, 40% z drugiej grupy, 20% z trzeciej grupy. Celem zastosowania drugiej metody jest utrzymanie różnorodności jak i dobrej jakości rozwiązań.

Na rysunku 4 przedstawiony jest schemat blokowy zastosowanego algorytmu genetycznego.

5. WYNIKI BADAŃ

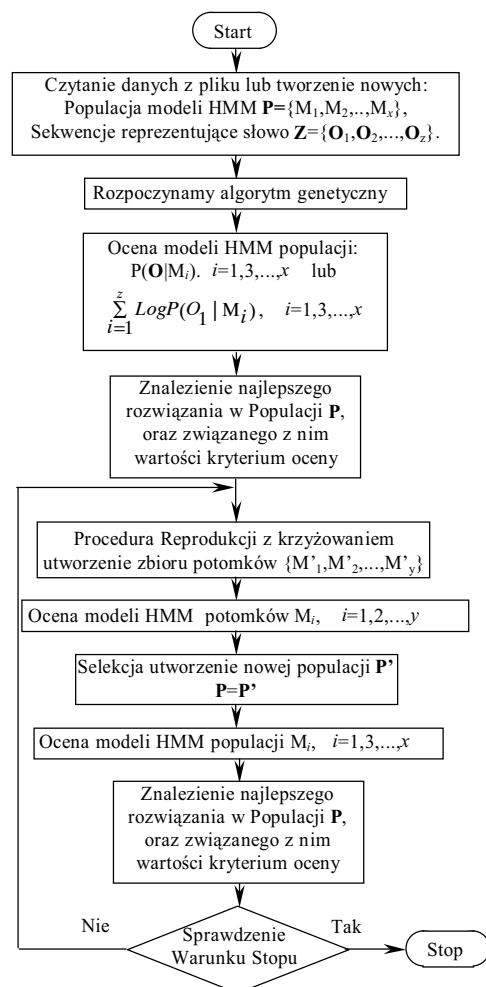
W niniejszym rozdziale przedstawione są wyniki badań skuteczności podanych w rozdziałach 2 oraz 3 algorytmów optymalizacji modeli HMM. Wyniki badań podane są dla sześciu sylab (sekwencji uczących), wyodrębnionych z nagrzanego dźwięku.

Sekwencje zostały podzielone na dwa zbiory:

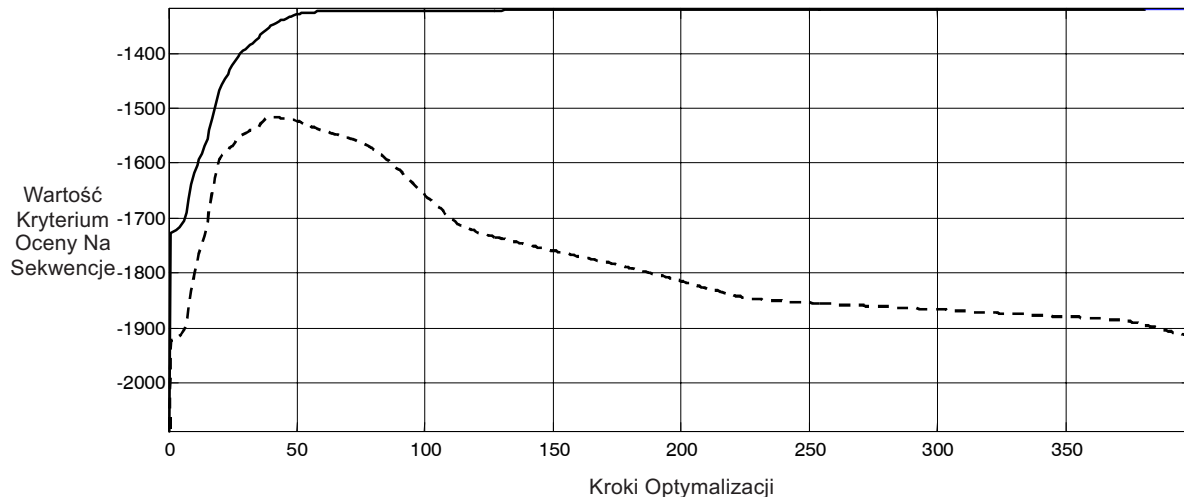
- 1) zbiór uczący, do którego trenowano modele HMM;
- 2) zbiór oceny.

Metody utworzenia sekwencji podane są w rozdziale 1. Wykorzystane w badaniach modele HMM miały po 10 węzłów.

W algorytmie standardowym stosowano zależności (31)–(33) estymacji ze skalowaniem współczynników $\alpha_t(i)$, $\beta_t(i)$. Optymalizacja była prowadzona przez 30 iteracji. Ponieważ podczas dłuższego procesu uczenia następuje przeuczenie modeli HMM, modele bardzo dobrze reprezentują wersje sylab, do których były uczone, natomiast pogarsza się reprezentacja sylab rozpoznawanych. Efekty przeuczenia można zaobserwować na rysunku 5. Ciągłą linią zaznaczone jest kryterium oceny dla zbioru uczącego sylab, natomiast przerywaną linią zaznaczone jest kryterium oceny dla



Rys. 4. Schemat blokowy algorytmu genetycznego zastosowanego do optymalizacji modeli HMM



Rys. 5. Efekty przeuczenia modelu HMM. Linia ciągła reprezentuje wartość kryterium oceny dla zbioru uczącego, natomiast linia przerywana kryterium oceny dla zbioru oceny. Zbiór uczący był złożony z pięciu sekwencji odpowiadających sylabie „cja”, ocena została przeprowadzana do jednej sekwencji sylaby „cja”

zbioru oceny. Jak widać z rysunku 5, po 40 iteracjach występują maksymalne wartości kryterium dla zbioru oceny.

Po zastosowaniu algorytmu Bauma–Welcha uzyskano modele HMM odpowiadające poszczególnym sylabom. W tabeli 1 przedstawione są wartości kryterium oceny modeli na sekwencję dla zbiorów sekwencji odpowiadających danej sylabie. Wartość kryterium zostały wyznaczone dla modeli HMM trenowanych do tego zbioru sekwencji, do którego podana jest wartość kryterium.

W tabeli 2 pokazane są wartości prawdopodobieństwa wygenerowania sekwencji na $K = 600$ próbek, tj: $P(O_i | M)^{K/T_i}$. Prawdopodobieństwo to jest wyznaczane dla sekwencji, do których były trenowane modele HMM (pięć pierwszych wierszy), oraz dla sekwencji oceny (szósty wiersz). W tabeli 1 znajdują się sumy logarytmów wartości z kolumn tabeli 2 dzielone przez liczbę sekwencji treningowych. Wartości w tabeli 1, np. dla sylaby „cja” wyznaczone są z tabeli 2 (zaciemnione pola).

W tabelach 3 i 4 na przekątnej pokazane są prawdopodobieństwa na $K = 600$ próbek wygenerowania sekwencji przez modele HMM, do których były one trenowane (dla pierwszych dwóch nagrań). Na przekątnej tabeli 3 występują te same wartości, co w pierwszym rzędzie tabeli 2, na przekątnej tabeli 4 wartości z drugiego rzędu tabeli 2. Poza przekątnymi w tabelach 3 i 4 znajdują się wartości prawdopodobieństw wygenerowania sylab przez modele HMM, do których nie były one trenowane. W tabeli 5 pokazane są prawdopodobieństwa dla sekwencji oceny, stąd na przekątnej tabeli 5 występują te same wartości, co w szóstym rzędzie tabeli 2. Zaciemnionymi polami w tabelach 3, 4 i 5 zaznaczone są minimalne wartości w rzędach. Jak widać z tabeli 3, pomimo że model był trenowany do jakiejś sylaby, zdarza się, że modeluje on lepiej inną sylabę. W przypadku rozpoznawania sylab zostały uzyskane lepsze wyniki, zazwyczaj nie zdarzają się błędy rozpoznawania.

Tabela 1. Tabela wartości kryterium oceny modeli HMM, zależność (30), po optymalizacji. Modele optymalizowane za pomocą algorytmu Bauma–Welcha

	Model „cja” Sylaby „cja”	Model „li” Sylaby „li”	Model „ma” Sylaby „ma”	Model „op” Sylaby „op”	Model „ty” Sylaby „ty”	Model „za” Sylaby „za”
$\frac{\sum_{i=1}^5 \text{Log}P_i}{T_i} \cdot 600$ 5	-1,1524e+03	-1,4610e+03	-1,1059e+03	-7,4720e+02	-5,6543e+02	-1,3483e+03

Tabela 2. Tabela prawdopodobieństw wygenerowania sylab, do których były trenowane modele HMM. Sylaby utworzone ze słowa „optymalizacja”. Modele optymalizowane za pomocą algorytmu Bauma–Welcha

Nagranie	Model „cja” Sylaba „cja”	Model „li” Sylaba „li”	Model „ma” Sylaba „ma”	Model „op” Sylaba „op”	Model „ty” Sylaba „ty”	Model „za” Sylaba „za”
Uczące 1	9,7598e-475	2,1584e-633	4,8148e-596	3,1398e-604	3,1901e-649	1,4057e-641
Uczące 2	2,0063e-561	9,4854e-626	8,3522e-533	2,8516e-608	4,6065e-587	1,3776e-560
Uczące 3	1,8825e-590	1,3188e-646	3,7888e-526	7,3955e-668	3,9678e-663	1,2598e-549
Uczące 4	4,3433e-434	2,2326e-645	5,8510e-685	7,0160e-729	6,6833e-582	5,2015e-597
Uczące 5	2,3208e-445	3,3090e-625	1,0844e-665	1,8462e-639	7,1778e-592	8,6020e-583
Oceny 6	6,4696e-503	8,6507e-655	6,5226e-661	1,0800e-626	3,3718e-657	7,7907e-665

Tabela 3. Tabela dla nagrania uczącego 1 pokazująca, która sylaba z wypowiedzi słowa „optymalizacja” jest najlepiej modelowana przez poszczególne modele. Modele dobierane przez algorytm Bauma–Welcha

Model	Sylaba „cja”	Sylaba „li”	Sylaba „ma”	Sylaba „op”	Sylaba „ty”	Sylaba „za”
Sylaby „cja”	1,7e-475	6,9e-706	1,1e-725	3,3e-635	1,0e-710	3,5e-667
Sylaby „li”	2,2e-848	2,1e-633	5,3e-878	7,8e-1298	3,7e-685	9,7e-1225
Sylaby „ma”	9,8e-599	4,7e-766	4,8e-596	1,3e-667	2,0e-794	5,9e-732
Sylaby „op”	4,9e-774	8,3e-1048	5,6e-1019	3,1e-604	1,5e-1038	5,0e-869
Sylaby „ty”	2,4e-1096	1,1e-721	4,3e-1279	2,5e-1596	3,1e-649	6,6e-1317
Sylaby „za”	5,6e-539	1,2e-791	1,6e-723	1,5e-695	1,9e-825	1,4e-641

Kolorem szarym oznaczono minimalną wartość w rzędzie

Tabela 4. Tabela dla nagrania uczącego 2 pokazująca, która sylaba z wypowiedzi słowa „optymalizacja” jest najlepiej modelowana przez poszczególne modele. Modele dobierane przez algorytm Bauma–Welcha

Model	Sylaba „cja”	Sylaba „li”	Sylaba „ma”	Sylaba „op”	Sylaba „ty”	Sylaba „za”
Sylaby „cja”	2,0e-561	2,7e-695	1,7e-646	2,5e-702	5,0e-645	1,0e-719
Sylaby „li”	1,8e-945	9,4e-626	1,0e-904	1,8e-1509	6,8e-673	2,8e-1190
Sylaby „ma”	1,7e-713	1,7e-693	8,3e-533	1,2e-798	3,4e-698	8,8e-621
Sylaby „op”	1,6e-1020	2,3e-1163	6,0e-1008	2,8e-608	1,7e-810	6,7e-746
Sylaby „ty”	9,8e-1233	9,7e-853	6,4e-1204	5,0e-1750	4,6e-587	1,9e-1284
Sylaby „za”	1,3e-662	2,2e-765	1,2e-577	1,6e-714	5,3e-641	1,3e-560

Kolorem szarym oznaczono minimalną wartość w rzędzie

Tabela 5. Tabela dla nagrania oceny 6 pokazująca, która sylaba z wypowiedzi słowa „optymalizacja” jest najlepiej modelowana przez poszczególne modele. Modele dobierane przez algorytm Bauma–Welcha

Model	Sylaba „cja”	Sylaba „li”	Sylaba „ma”	Sylaba „op”	Sylaba „ty”	Sylaba „za”
Sylaby „cja”	6,4e-503	8,1e-746	8,5e-727	6,2e-797	5,1e-739	4,2e-684
Sylaby „li”	2,6e-801	8,6e-655	1,2e-1075	5,4e-1200	1,3e-748	1,1e-1660
Sylaby „ma”	1,1e-604	1,4e-673	6,5e-661	4,8e-738	1,2e-799	7,8e-856
Sylaby „op”	7,8e-760	2,5e-1231	5,0e-1663	1,0e-626	1,6e-925	1,2e-1115
Sylaby „ty”	3,1e-1156	1,3e-801	6,4e-1771	1,5e-1500	3,3e-657	2,3e-1947
Sylaby „za”	6,1e-502	6,1e-770	1,6e-742	1,5e-838	5,9e-784	7,7e-665

Kolorem szarym oznaczono minimalną wartość w rzędzie

Oznaczając minimalne wartości w kolumnach tabel 3, 4 i 5, tj: w przypadku rozpoznawania sylab, można stwierdzić, że zazwyczaj wybierane są właściwe modele HMM zarówno dla zbioru uczącego, jak i dla zbioru oceny. W przypadku zwiększania liczby iteracji algorytmu Bauma–Welch, liczba błędów dla zbioru uczącego maleje i po około 1000 iteracjach błędy są zerowe, zarówno w przypadku modelowania sylab, jak i rozpoznawania, natomiast zwiększa się liczba błędów dla zbioru oceny. Zwiększając liczbę sekwencji uczących, możemy jednocześnie zwiększyć liczbę iteracji algorytmu, nie wywołując efektów przeuczenia modeli.

Błędy modelowania sylaby „cja” z nagrania pierwszego spowodowane są tym, że w sylabie tej litera „c” ma bardzo niski poziom, co w wyniku dyskretyzacji powoduje występowanie w sekwencji uczącej wielu wartości zerowych na poziomie 16. Dlatego dla tej sekwencji uzyskuje się większe wartości prawdopodobieństwa.

Zaproponowany algorytm genetyczny zastosowany do losowych rozwiązań początkowych działa wolniej od algorytmu Bauma–Welcha oraz daje gorsze rozwiązania, co można było przewidzieć.

Jeżeli jako kryterium optymalizacji modeli HMM zostanie przyjęta wartość funkcji oceny modelu, równanie (30), wówczas algorytm genetyczny może zostać wykorzystany do wychodzenia z optimum lokalnych, które zostały znalezione dzięki algorytmowi Bauma–Welcha. Badania algorytmu genetycznego zostały przeprowadzone dla modeli HMM o pięciu węzłach. Model został trenowany do dwóch sekwencji odpowiadających sylabie „cja”. Początkowa wartość kryterium oceny dla modelu wynosiła $2,026065446411371194e+03$. W wyniku optymalizacji przez 3500 iteracji zostało uzyskane optimum lokalne, w którym wartość funkcji oceny wynosiła $-1,435104751597842861e+03$ i nie następowała dalsza poprawa rozwiązania. Dla algorytmu genetycznego została utworzona populacja zawierająca uzyskane dzięki algorytm-

mowi Bauma–Welcha rozwiązanie oraz rozwiązania losowe. W wyniku działania algorytmu genetycznego przez 50 iteracji uzyskano zmodyfikowane losowo rozwiązanie, które ponownie zostało poddane optymalizacji algorytmem Bauma–Welcha przez 800 iteracji. Zostało znalezione rozwiązanie lokalne optymalne o wartości funkcji oceny – $1,270414874341903282e+03$ i nie następowała dalsza poprawa znalezionej wartości.

6. WNIOSKI

Zaproponowany algorytm genetyczny zastosowany do losowych rozwiązań początkowych działa wolniej od algorytmu Bauma–Welcha oraz daje gorsze rozwiązania, co można było przewidzieć. Zaproponowany algorytm genetyczny może nawet doprowadzić do pogorszenia wartości rozwiązania. Pogorszenie rozwiązania następuje, jeżeli rozwiązanie nowej populacji wybierane są zgodnie z zastosowaną metodą (*Elite Group*), w przypadku wyboru najlepszych rozwiązań nie następuje modyfikacja najlepszego wcześniej znalezionej rozwiązania.

Dzięki zaproponowanemu algorytmowi genetycznemu nie udało się uzyskać poprawy wartości znalezionej przez algorytm Bauma–Welcha rozwiązania, ale uzyskano efekt wyjścia z optimum lokalnego. Po kilku krokach algorytmu genetycznego uzyskane dzięki niemu rozwiązanie, pomimo że o gorszej wartości funkcji oceny, ponownie zostało poddane optymalizacji algorytmem Bauma–Welcha, w wyniku czego nastąpiło znalezienie lepszych wartości rozwiązań niż wartość poprzednio znalezionej maksimum. Jednak wydaje się, że bardziej skutecznym rozwiązaniem powinno być zastosowanie algorytmu Bauma–Welcha do kilku losowych rozwiązań początkowych, a następnie wybranie spośród nich rozwiązania najlepszego.

Literatura

- [1] MacDonald I.L., Zucchini W.: *Hidden Markov and Other Models for Discrete-valued Time Series*. London, Chapman & Hall 1997, ISBN 0-412-55850-5
- [2] Juang B.H., and Rabiner L.R.: *Hidden Markov models for speech recognition*. *Technometrics*, 33(3), 1991, 251–272
- [3] Shomali M., Kapusta M., Gajer M.: *Zastosowanie niejawnych modeli Markowa w systemach automatycznego rozpoznawania mowy*. *Kwartalnik AGH Elektrotechnika i Elektronika*, t. 18, z. 3, 1999, s. 89–98
- [4] Gąciarz T.: *Hidden Markov Model (HMM) – opis modelu i algorytmów pod kątem wykorzystania w problemach rozpoznawania mowy i pisma*. *Elektrotechnika*, t. 17, z. 1, 1998, s. 17–31
- [5] Iosifescu M.: *Skończone procesy Markowa i ich zastosowania*. Warszawa, PWN 1988
- [6] In-Chan Choi, Seong-In Kim, Hak-Soo Kim.: *A genetic Algorithm with a mixed region search for the asymmetric traveling salesman problem*. *Computers & Operations Research*, 30, 2003, 773–786
- [7] Barrie M. Baker, M.A Ayechev.: *A genetic algorithm for the vehicle routing problem*. *Computers & Operations Research*, 30, 2003, 787–800.
- [8] Karp R.: *Patching algorithm for nonsymmetric traveling-salesman problem*. *SIAM Journal Computing*, 8, 1979, 561–73
- [9] Bellmore M., Malone J.: *Patology of traveling-salesman subtour-elimination algorithms*. *Operations Research*, 19, 1971, 278–307
- [10] Lin S., Kernighan B.: *An effective heuristic algorithm for the traveling salesman problem*. *Operation Research*, 21, 1973, 498–516
- [11] Levinson S.E., Rabiner L.R., Sandhi M.M.: *An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition*. *Bell System Tech. J.*, 62, April 1983, 4, 1035–1074

Wpłynęło: 30.03.2005

Karol SZOSTEK



Urodził się 22.05.1974 roku Ukończył kierunek automatyka na Wydziale Elektrotechniki, Automatyki Informatyki i Elektroniki Akademii Górniczo-Hutniczej im. Stanisława Staszica w Krakowie. Obecnie jest słuchaczem III roku Studium Doktoranckiego na AGH. Prowadzi badania w dziedzinie modelowania probabilistycznego oraz automatycznego rozpoznawania mowy.

e-mail: kszostek@galaxy.uci.agh.edu.pl