NHẬN DẠNG TIẾNG NÓI 10 CHỮ SỐ VIỆT NAM PHÁT ÂM RỜI BẰNG MÔ HÌNH MARKOV ẨN

Trần Tiến Đức
Trường Đại Học Sư Phạm Kỹ Thuật
Hoàng Kiếm
Trường Đại Học Khoa Học Tự Nhiên
(Bài nhận ngày 16/03/1999)

TÓM TẮT : Bài báo này nghiên cứu một phương pháp hiệu quả dùng để mô hình hóa cấu trúc động của tiếng nói là mô hình Markov ẩn. Đây là hướng tiếp cận đối sánh mẫu xác suất, ở đó các mẫu tiếng nói tuần tự theo thời gian là kết quả của quá trình thống kê hay ngẫu nhiên. Trong tiếng nói, mỗi từ được chia thành nhiều frame dài 10-30ms, với mỗi frame chúng ta trích ra một tập các hệ số gọi là các hệ số dự báo tuyến tính - linear predictive coefficients (LPC) rồi chuyển các hệ số này thành vector cepstral. Thông qua lượng tử hóa vector, các vector đặc tính này được biến đổi thành các quan sát và dùng mô hình Makov ẩn - hidden Markov model (HMM) để ước lượng và nhận dạng theo các thuật toán của Baum-Welch. Phần kết quả cho thấy chúng ta nên sử dụng HMM để nhận dạng tiếng nói Việt Nam.

1. GIỚI THIỆU HMM

Trong tiếng nói, mỗi từ được chia thành nhiều frame, rồi dùng phương pháp dãy bộ lọc hay LPC cepstral để trích đặc tính của từng frame này, mỗi frame được trích thành một vector [1][2][3]. Kết quả chúng ta có được các vector đặc tính của một từ và ký hiệu là $\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_t, ... \mathbf{t}_T$. Thông qua lượng tử hóa vector, các vector đặc tính này được biến đổi thành các quan sát và ký hiệu là $\mathbf{o}_1, \mathbf{o}_2, ..., \mathbf{o}_t, ..., \mathbf{o}_T$ theo ngôn ngữ của HMM. Chú ý rằng chúng ta đã ký hiệu lại chỉ số quan sát là t (i dành cho mục đích khác) và số lượng quan sát là T.

Hình 1 là mô hình Markov ẩn gồm một xích Markov. Mỗi vòng tròn biểu diễn một trạng thái của mô hình và ở thời điểm rời rạc t, tương ứng với một frame của tiếng nói, mô hình sẽ ở một trong những trạng thái này và tạo ra một mẫu tiếng nói hay một quan sát. Ở thời điểm t+1, mô hình sẽ di chuyển đến trạng thái mới hay vẫn ở trạng thái cũ và tạo ra một mẫu khác. Lặp lại quá trình này cho đến khi tạo ra toàn bộ các mẫu. Sở dĩ được gọi là 'ẩn' bởi vì không thể xác định được các trạng thái tạo ra tương ứng với các quan sát đã cho.

2. ĐÁNH GIÁ XÁC SUẤT

Để tính xác suất của các quan sát $\mathbf{O} = (\mathbf{o}_1, \mathbf{o}_2, ..., \mathbf{o}_T)$ với mô hình $\lambda = (A, B, \pi)$, tức là muốn tính $P(\mathbf{O}|\lambda)$. Chúng ta sử dụng thuật toán Forward hay Baum-Welch [1][2]. Khảo sát biến thuận forward $\alpha_t(i)$ được định nghĩa như sau:

$$\alpha_t(i) = P(\mathbf{o}_1\mathbf{o}_2...\mathbf{o}_t, q_t = i|\lambda) \quad (1)$$

tức là xác suất của miền quan sát $\mathbf{o}_1\mathbf{o}_2...\mathbf{o}_t$ (đến thời điểm t) và trạng thái i ở thời điểm t, ứng với mô hình λ. Có thể tính $\alpha_t(i)$ bằng qui nạp như sau:

1. Khởi tạo

$$\alpha_1(i) = \pi_i b_i(\mathbf{o}_1) \text{ với } 1 \le i \le N \quad (2)$$

2. Qui nạp

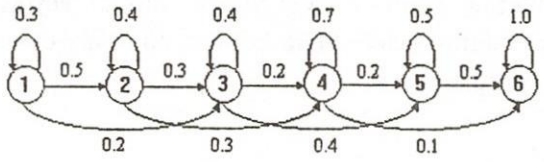
$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(\mathbf{o}_{t+1})$$
$$1 \le t \le T-1$$
$$1 \le j \le N \quad (3)$$

3. Kết thúc

$$P(\mathbf{O}|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \quad (4)$$

$$A = [a_{ij}] = \begin{bmatrix} 0.3 & 0.5 & 0.2 & 0 & 0 & 0 \\ 0 & 0.4 & 0.3 & 0.3 & 0 & 0 \\ 0 & 0 & 0.4 & 0.2 & 0.4 & 0 \\ 0 & 0 & 0 & 0.7 & 0.2 & 0.1 \\ 0 & 0 & 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 1.0 \end{bmatrix}$$

$\pi = (0.5,0.5,0,0,0,0)$



$$B = [b_{jk}] = \begin{bmatrix} b_{11} & b_{21} & b_{31} & b_{41} & b_{51} & b_{61} \\ b_{12} & b_{22} & b_{32} & b_{42} & b_{52} & b_{62} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ b_{1k} & b_{2k} & b_{3k} & b_{4k} & b_{5k} & b_{6k} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ b_{1M} & b_{2M} & b_{3M} & b_{4M} & b_{5M} & b_{6M} \end{bmatrix}$$

Hình 1 Ví dụ về mô hình Markov ẩn.

Tương tự, định nghĩa biến ngược backward

$$\beta_t(i) = P(\mathbf{o}_{t+1}\mathbf{o}_{t+2}\ldots\mathbf{o}_T|q_T = i,\lambda) \quad (5)$$

tức là xác suất của miền quan sát từ $t+1$ đến cuối T, với trạng thái i ở thời điểm t, ứng với mô hình λ. Cũng tính $\beta_t(i)$ bằng qui nạp:

1. Khởi tạo

$$\beta_T(i) = 1 \quad 1 \le i \le N \quad (6)$$

2. Qui nạp

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(\mathbf{o}_{t+1}) \beta_{t+1}(j)$$
$$t = T-1, T-2, \ldots, 1 \quad 1 \le i \le N \quad (7)$$

3. Ước lượng tham số

Trong phần này chúng ta sẽ khảo sát phương pháp lặp do Baum và các cộng sự đề xuất để chọn tham số của mô hình sao cho thích hợp nhất - maximum likelihood (ML) [1][2].

$$\overline{\pi}_i = \frac{\alpha_1(i)\beta_1(i)}{\sum_{i=1}^{N}\alpha_T(i)} \quad (8)$$

$$\overline{a}_{ij} = \frac{\sum_{t=1}^{T-1}\alpha_t(i)a_{ij}b_j(\mathbf{o}_{t+1})\beta_{t+1}(j)}{\sum_{t=1}^{T-1}\alpha_t(i)\beta_t(i)} \quad (9)$$

$$\overline{b}_{jk} = \frac{\sum_{t=1}^{T}\alpha_t(j)\beta_t(j)\delta(\mathbf{o}_t,\mathbf{v}_k)}{\sum_{t=1}^{T}\alpha_t(j)\beta_t(j)} \quad (10)$$

ở đó $\delta(\mathbf{o}_t,\mathbf{v}_k)=1$ nếu $\mathbf{o}_t=\mathbf{v}_k$

$=0$ ngược lại

Nếu gọi mô hình hiện tại là $\lambda=(A,B,\pi)$ rồi dùng mô hình hiện tại này để tính vế phải của các Phương trình (8), (9) và (10), chúng ta sẽ có được mô hình $\overline{\lambda}=(\overline{A},\overline{B},\overline{\pi})$. Baum và cộng sự đã chứng minh rằng:

1. Mô hình λ đạt tới trạng thái tới hạn nếu $\overline{\lambda}=\lambda$.
2. Nếu mô hình $\overline{\lambda}$ tốt hơn mô hình λ thì $P(\mathbf{O}|\overline{\lambda})>P(\mathbf{O}|\lambda)$.

Tức là đã chọn được mô hình mới $\overline{\lambda}$ tốt hơn ứng với quan sát đã cho.

4.VẤN ĐỀ TỶ LỆ KHI TỔ CHỨC MÔ HÌNH MARKOV ẨN

Trong công thức đệ qui của thuật toán Baum-Welch, chúng ta tích lũy tích các xác suất ở mỗi thời điểm t, vì vậy $\alpha_t(i)$ nhanh chóng tiến dần về zero. Với t đủ lớn (≥ 100), phạm vi của $\alpha_t(i)$ sẽ vượt quá sự cho phép của bất kỳ máy nào (ngay cả kiểu chính xác kép). Vì vậy cần đưa hệ số tỷ lệ vào thuật toán đánh giá xác suất và ước lượng.

Cơ bản của phương pháp tỷ lệ là nhân $\alpha_t(i)$ với hệ số tỷ lệ, hệ số tỷ lệ này độc lập với i (tức là chỉ phụ thuộc vào t) với mục đích là giữ cho $\alpha_t(i)$ đã tỷ lệ nằm trong phạm vi cho phép của máy tính. Phương pháp tỷ lệ cũng thực hiện tương tự cho các số $\beta_t(i)$ (bởi vì các số này cũng tiến nhanh tới zero). Ở cuối quá trình tính toán, các hệ số tỷ lệ sẽ được gỡ bỏ để thu được $\alpha_t(i)$ [1][2].

Để hiểu phương pháp tỷ lệ rõ hơn, chúng ta hãy khảo sát quá trình tính toán $\alpha_t(i)$. Ký hiệu $\alpha_t(i)$ là α chưa tỷ lệ, $\hat{\alpha}_t(i)$ là α đã tỷ lệ, và $\hat{\hat{\alpha}}_t(i)$ là biến trung gian của α trước khi tỷ lệ. Khởi tạo, với t=1, tính $\alpha_1(i)$ theo Phương trình (2) và đặt $\hat{\hat{\alpha}}_1(i)=\alpha_1(i)$, $c_1=\frac{1}{\sum_{i=1}^{N}\alpha_1(i)}$ và $\hat{\alpha}_1(i)=c_1\alpha_1(i)$. Với mỗi t, $2\leq t\leq T$, đầu tiên tính $\hat{\hat{\alpha}}_t(i)$ theo công thức qui nạp ở Phương trình (3) theo biến đã tỷ lệ $\hat{\alpha}_t(i)$, tức là

$$\hat{\hat{\alpha}}_t(i) = \sum_{j=1}^{N} \hat{\alpha}_{t-1}(j) a_{ji} b_i(\mathbf{o}_t) \quad (11)$$

Xác định hệ số tỷ lệ c_t

$$c_t = \frac{1}{\sum_{i=1}^{N} \hat{\hat{\alpha}}_t(i)} \quad (12)$$

và

$$\hat{\alpha}_t(i) = c_t \hat{\hat{\alpha}}_t(i) \quad (13)$$

hay

$$\hat{\alpha}_t(i) = \frac{\sum_{j=1}^{N} \hat{\alpha}_{t+1}(j) a_{ji} b_i(\mathbf{o}_t)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \hat{\alpha}_{t-1}(j) a_{ji} b_i(\mathbf{o}_t)} \quad (14)$$

Bằng qui nạp, chúng ta có thể viết

$$\hat{\alpha}_{t-1}(j) = \left(\prod_{\tau=1}^{t-1} c_\tau \right) \alpha_{t-1}(j) \quad (15)$$

Như vậy

$$\hat{\alpha}_t(i) = \frac{\sum_{j=1}^{N} \alpha_{t-1}(j) \left(\prod_{\tau=1}^{t-1} c_\tau \right) a_{ji} b_i(\mathbf{o}_t)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_{t-1}(j) \left(\prod_{\tau=1}^{t-1} c_\tau \right) a_{ji} b_i(\mathbf{o}_t)}$$

$$= \frac{\alpha_t(i)}{\sum_{i=1}^{N} \alpha_t(i)} \quad (16)$$

tức là hệ số tỷ lệ của $\alpha_t(i)$ là tổng các trạng thái của $\alpha_t(i)$.

Tương tự, chúng ta xác định hệ số tỷ lệ cho $\beta_t(i)$

$$\hat{\beta}_{t+1}(j) = \left[\prod_{s=t+1}^{T} c_s \right] \beta_{t+1}(j) \quad (17)$$

Chú ý rằng hệ số tỷ lệ c_t cũng được dùng để tỷ lệ $\beta_t(i)$

Bởi vì hệ số tỷ lệ làm cho biên độ của α tiến đến 1, và bởi vì biên độ của α và β là so sánh được, nên sử dụng cùng hệ số tỷ lệ cho α và β là cách hiệu quả để việc tính toán nằm trong phạm vi cho phép của máy tính.

Dựa vào Phương trình (15) và (17), chúng ta suy ra công thức ước lượng ở Phương trình (9) theo biến thuận forward và biến ngược backward đã tỷ lệ là

$$\overline{a}_{ij} = \frac{\sum_{t=1}^{T-1} \hat{\alpha}_t(i) a_{ij} b_j(\mathbf{o}_{t+1}) \hat{\beta}_{t+1}(j)}{\sum_{i=1}^{T-1} \sum_{j=1}^{N} \hat{\alpha}_t(i) a_{ij} b_j(\mathbf{o}_{t+1}) \hat{\beta}_{t+1}(j)} \quad (18)$$

Chú ý phương pháp tỷ lệ ở trên cũng áp dụng tốt cho công thức ước lượng các hệ số của B và π.

Điều thực sự thay đổi trong HMM là tính xác suất $P(\mathbf{O}|\lambda)$. Không thể tổng các số hạng $\hat{\alpha}_T(i)$ bởi vì những số hạng này đã bị tỷ lệ. Tuy nhiên, dựa vào Phương trình (15), chúng ta có tính chất sau đây:

$$\prod_{t=1}^{T} c_t \sum_{i=1}^{N} \alpha_T(i) = C_T \sum_{i=1}^{N} \alpha_T(i) = 1 \quad (19)$$

Như vậy

$$\prod_{t=1}^{T} c_t P(\mathbf{O}|\lambda) = 1 \quad (20)$$

$$P(\mathbf{O}|\lambda) = \frac{1}{\prod_{t=1}^{T} c_t} \quad (21)$$

$$\log[P(\mathbf{O}|\lambda)] = -\sum_{t=1}^{T} \log c_t \quad (22)$$

Như vậy log của P được tính chứ không phải P, bởi vì P có khả năng vượt quá phạm vi cho phép của máy tính.

5. TỔ CHỨC NHẬN DẠNG TỪ RỜI BẰNG MÔ HÌNH MARKOV ẨN

Giả sử cần nhận dạng bộ từ vựng có V từ, mỗi từ đều có HMM riêng và được nói K lần (có thể một hay nhiều người nói), chúng ta cần thực hiện các bước sau đây:

1. Với mỗi từ v trong bộ từ vựng, phải xây dựng một mô hình Markov ẩn λ_v, tức là phải ước lượng các tham số của mô hình (A, B, π) sao cho tối ưu likelihood dựa trên tập dữ liệu huấn luyện. Mỗi từ được đọc 100 lần, rồi dùng thuật toán ước lượng để xác định (A, B, π) của từ đó.

2. Để nhận dạng từ chưa biết, chúng ta chia từ đó thành nhiều frame, trích đặc điểm bằng phương pháp LPC, rồi chuyển thành các hệ số cepstral, thông qua bộ lượng tử hóa vector chúng ta có được quan sát $\mathbf{O} = \{\mathbf{o}_1 \mathbf{o}_2 \ldots \mathbf{o}_T\}$. Tiếp theo tính xác suất cho tất cả mô hình $P(\mathbf{O}|\lambda_v)$, $1 \le v \le V$, và chọn từ có xác suất lớn nhất, tức là

$$v^* = \arg \max_{1 \le v \le V} [P(\mathbf{O}|\lambda_v)] \quad (23)$$

6. Kết luận

Chúng tôi đã cài đặt chương trình nhận dạng phụ thuộc người nói 10 chữ số Việt Nam phát âm rời và rõ ràng bằng Borland C++ 4.5 for Windows. Tiếng nói được lấy mẫu ở tần số 11025Hz. Mỗi chữ số được đọc 100 lần. Mỗi frame dài 128 bytes, phủ lấp lên nhau 64 bytes. Vector đặc tính là 12 hệ số dự báo tuyến tính, rồi chuyển thành 24 hệ số cepstral. Lượng tử hóa vector có kích thước codebook $M = 64$ với độ đo cepstral cắt, mô hình Markov ẩn là mô hình Bakis - bước nhảy hai, có số trạng thái là 4. Thời gian lượng tử hóa vector mất 10 giờ, thời gian học mỗi chữ số mất 1 giờ, như vậy 10 chữ số mất 10 giờ, tổng cộng mất chừng 20 giờ trên máy vi tính Pentium - 166mHz, 64mB RAM.

Để đánh giá độ chính xác của chương trình, chúng tôi đã dùng một cuốn danh bạ điện thoại để đọc các số điện thoại ngẫu nhiên, mỗi chữ số được đọc 100 lần. Kết quả kiểm tra trình bày ở Bảng 1.

Bảng 1 Kết quả kiểm tra.

	0	1	2	3	4	5	6	7	8	9
0	100									
1		94			6					
2			93					6	1	
3				100						
4		2			98					
5						100				
6							100			
7								100		
8									100	
9						2		1		97

Ngày nay, mô hình Markov ẩn đã được ứng dụng rộng rãi trên thế giới để nhận dạng tiếng nói như các hệ Naturally Speaking Deluxe của Dragon Systems, Via Voice của IBM, Kurzweil VoicePro của Lernout & Hauspie.... Tuy nhiên việc ứng dụng mô hình Markov ẩn vào tiếng Việt hãy còn đang nghiên cứu ở Việt Nam, ngoài ra tiếng Việt là tiếng đơn âm có dấu nên phần trích đặc điểm có phần khác biệt so với tiếng Anh.

7. Hướng phát triển

Trong bài báo này, chúng tôi chỉ đề cập đến phương pháp LPC-cepstral để trích đặc điểm tiếng nói của chữ số, trong tương lai có thể dùng phương pháp biến đổi wavelet để trích đặc điểm và thanh điệu (dấu) của tiếng nói Việt Nam [6].

Cũng không thể không thảo luận tới việc nhận dạng từ nối, tức là những từ phát âm liền nhau và thực sự không thể tách rời như phát âm số điện thoại bình thường. Các thuật toán lập trình động hai mức, tạo mức hay một lần duyệt (đồng bộ frame) dùng để giải quyết vấn đề này.

Với bộ từ vựng không hạn chế số lượng từ, việc xây dựng mô hình Markov ẩn cho từng từ là không chấp nhận được. Vì vậy cần phải chia một từ thành các từ con, xây dựng mô hình Markov ẩn cho các từ con, nhận dạng các từ con đó, kết hợp các từ con dựa vào xác suất và ngữ pháp, ngữ nghĩa để xác định một từ rồi một câu.

RECOGNITION OF THE VIETNAMESE SPEECH 'S ISOLATED DIGITS USING HIDDEN MARKOV MODEL

Tran Tien Duc – Hoang Kiem

ABSTRACT : A powerful technique for modelling the temporal structure and variability in speech is one called hidden Markov modelling. This is a probabilistic pattern-matching approach that models a time-sequence of speech patterns as output of a stochastic or random process. In

speech processing, a word is analyzed in 10-30ms time frames and for each frame, we extract a set of coefficients which is called linear predictive coefficients (LPC) and convert one to a cepstral vector. Through a vector quantizer, the vector is mapped to a observation and using hidden Markov model (HMM) to estimate and recognise the observation by Baum-Welch algorithms. The results show that we should used the HMM to recognise Vietnamese speech.

TÀI LIỆU THAM KHẢO

[1] L. Rabiner and B. H. Juang, Fundamentals of speech recognition, Published by PTR Prentice-Hall, Englewood Cliffs, New Jersey 07632, 1993.

[2] John R. Deller, Jr., John G. Proakis, John H. L. Hansen, Discrete-time processing of speech signals. Published by Macmillan, 1993.

[3] F. J. Owens, Signal processing of speech, Published by Macmillan, London, 1993.

[4] Daniel J. Mashao, Yoshihiko Gotoh, and Harvey F. Silverman, 'Analysis of LPC/DFT Features for an HMM-based Alphadigit Recognizer', IEEE Signal processing letters, Vol. 3, No. 4, April 1996.

[5] Hoàng Kiếm et al, Nhận dạng - các phương pháp và ứng dụng, Nhà xuất bản Thống kê, 1992.

[6] Thuong Le-Tien, 'A study on the continuous wavelet transform for the Vietnamese speech processing', Proceedings of the 97 international conference on natural information and intelligent information systems, Vol. 2, New Zealand, 1997.