Journal of Mechanics Engineering and Automation 10 (2020) 41-45
doi: 10.17265/2159-5275/2020.02.001

Application of Hidden Markov Models in Speech Command Recognition

Shing-Tai Pan, Zong-Hong Huang, Sheng-Syun Yuan, Xu-Yu Li, Yu-De Su and Jia-Hua Li

Department of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung 811, Taiwan, R.O.C.

Abstract: In this study, vector quantization and hidden Markov models were used to achieve speech command recognition. Pre-emphasis, a hamming window, and Mel-frequency cepstral coefficients were first adopted to obtain feature values. Subsequently, vector quantization and HMMs (hidden Markov models) were employed to achieve speech command recognition. The recorded speech length was three Chinese characters, which were used to test the method. Five phrases pronounced mixing various human voices were recorded and used to test the models. The recorded phrases were then used for speech command recognition to demonstrate whether the experiment results were satisfactory.

Key words: HMMs, Mel-frequency cepstral coefficients, speech command recognition, vector quantization.

1. Introductions

Following advances in modern technology, human-computer interaction (HCI) has become increasingly diversified. In the era of feature phones, physical keys were used. Nowadays, touchpads are used for smartphones. Movement and face recognition, eye control, and voice control will be used for HCI in the future. Among the aforementioned methods, voice control offers a minimum distance limitation and is particularly convenient to use at home. Some relative papers can be found in the papers [1-7] and the references therein.

In this study, vector quantization combined with Hidden Markov Models (HMMs) was used for speech command recognition. The characteristics of the state transition in HMMs can be applied in time series of voice (speech) data to achieve speech command recognition.

2. Voice Signal Preprocessing

This experiment featured a microphone to record and process sound. The sound was recorded as an analog signal. Therefore, for computation, the analog signal had to be converted into a digital signal. Voices produce different signals at different times, and not everyone's voice signal is the same. Therefore, voice signals must first be processed for voice or speech recognition. Once voice feature values are obtained, they can be used for recognition.

2.1 Extracting Voice Frames

Voice signals vary at different times. The division of a voice signal into short intervals is called a frame. The method for extracting a frame is to ensure that the end of a frame does not connect to the start of another frame. Frames overlap. The overlapping rate is typically 50%. A correct frame overlapping rate influences the feature values obtained. A high overlapping rate enhances the precision required for the extraction of feature values and requires more calculations, compared with a low overlapping rate.

2.2 Voice (Speech) Pre-emphasis

When voice (speech) is transmitted in the air, because of sound waves' characteristics, high-frequency signals are more markedly attenuated than low-frequency signals. Therefore, pre-emphasis

Corresponding author: Shing-Tai Pan, Ph.D., professor, research fields: speech recognition, intelligent signal processing, bio-signal processing and classification.

on high-frequency voice signals is necessary to balance the attenuation of the original signals. The original signals are input into a high-pass finite impulse response-based filter. The equation for the high-pass filter is as follows [8]:

$$S_{pe}(n) = S_{of}(n) - 0.97 \times S_{of}(n-1),\ 1 \le n \le N \qquad (1)$$

where $1 \le n \le N$; $S_{\text{pe}}(n)$ is the signal after pre-emphasis; $S_{\text{of}}(n)$ is the original signal; and N is the length of signal S.

2.3 Applying a Hamming Window

Because the start and the end of a frame are not necessarily identical, strong high-frequency waves may occur. Therefore, by applying a hamming window, continuity at the two ends of a frame can be maintained to reduce the occurrence of strong high-frequency waves. The equation for the hamming window is as follows [9]:

$$W(n) = \begin{cases} 0.54 - 0.46\cos(\frac{2n\pi}{N-1}),\ 0 \le n \le N-1; \\ 0, otherwise \end{cases} \qquad (2)$$

where N denotes the length of a frame, and n is the index of the frame.

$$F(n) = W(n) \times S(n); \qquad (3)$$

where $S(n)$ is the frame signal, and $F(n)$ is the voice signal after a hamming window is applied.

2.4 Fast Fourier Transform (FFT)

To extract the feature values, a frequency domain is required to process the voice signals. Originally, voice signals are presented using a time domain. Therefore, frames must undergo FFT, thus converting time domain signals into frequency domain signals. The corresponding equation is as follows [10]:

$$X_k = \sum_{n=0}^{N-1} S_w(n) \times W_N^{kn}, 0 \le k \le N-1 \qquad (4)$$

$$W_N = e^{\frac{-j2\pi}{N}} \qquad (5)$$

2.5 Mel-Frequency Cepstral Coefficients and Feature Values

Because the human auditory perception is nonlinear, and human ears are less sensitive to high-frequency signals than they are to low-frequency signals, Mel-frequency cepstrum is used to extract feature values and to simulate the reception of voice (speech) signals by human ears. First, frames undergo FFT. Second, the outcome is multiplied by a Mel triangular bandpass filter. The equation for the filter is as follows:

$$B_m(k) = \begin{cases} 0, k < f_{m-1} \\ \frac{k - f_{m-1}}{f_m - f_{m-1}}, f_{m-1} \le k \le f_m \\ \frac{f_{m+1} - k}{f_{m+1} - f_m}, f_m \le k \le f_{m+1} \\ 0, f_{m+1} < k \end{cases} \quad 1 \le m \le M \qquad (6)$$

where M denotes the number of filters. The logarithm of the summation can be obtained as follows:

$$Y(m) = log\left\{ \sum_{k=f_{m-1}}^{f_{m+1}} |X_k| B_m(k) \right\}. \qquad (7)$$

In the following equation, M denotes the number of $Y(m)$, and $Y(m)$ undergoes discrete cosine transform.

$$c_x(n) = \frac{1}{M} \sum_{m=1}^{M} Y(m) \times cos(\frac{\pi n(m - \frac{1}{2})}{M}) \qquad (8)$$

where $C_x(n)$ is the Mel-frequency cepstral parameter. To simplify the calculation, the first 13 pieces of data are used and then set as a feature vector for the following training and testing for HMM model.

3. HMMs

Once the framework of an artificial neural network (ANN) is determined, dynamic changes cannot be made, and only external processing or the same number of inputs can be adopted for analysis. Therefore, the integrity of the data can be influenced. Nevertheless, probability statistics for HMMs are more suitable than ANN for analyzing voice signals with variable lengths of feature vectors.

3.1 Vector Quantization

The observation (feature vectors) of the HMM results in a finite set, but 10-dimensional real vector space is infinite. Therefore, vector quantization is required [7]. To quantize each feature vector of the training samples, the distance d_k between a feature vector $v_f = [v_{f0} \ v_{f1} \ ... \ v_{fn}]^T$ and the *k*th vector (a row in the codebook) $V_{ck} = [V_{ck0} \ V_{ck1} \ ... \ V_{ckn}]^T$ in the codebook is calculated as follows:

$$d_k(v_f) = \sqrt{\sum_{i=0}^{n}(v_{fi} - V_{cki})^2} \tag{9}$$

Then, the feature vector v_f is classified to the group whose center (a row in the codebook) is with the smallest d_k. The row is then updated as follows.

$$\hat{V}_k = \frac{1}{N_k} \times \sum_{n=1}^{N_k} v_{fn}^k \tag{10}$$

in which $\hat{V}_k$ is the updated row (center vector of *k*th group), v_{fi}^k is the *i*th feature vector in *k*th group, N_k is the number of feature vectors in *k*th group. The codebook is trained by repeating the above steps until the codebook converges.

3.2 HMMs

An HMM yields a series of observation outcomes, and the series of state transitions is hidden. This structure can be applied to describe voice characteristics. The series of frames represent the observation outcomes. The parameters that require training are the probabilities of various observation outcomes for various states and the probabilities of various state transitions. Because the configured model starts at the first state, the probability of the initial state does not require training. Before describing the training method, a definition and introduction of HMM must be provided [11]:

$\lambda = \{A, B, \pi, S, V\}$: HMM;

$S = \{s_1, s_2, \ldots, s_N\}$: *N* states;

$V = \{v1, v2, ..., vM\}$: *M* observation outcomes;

$A = \{ a_{ij} \}, a_{ij} = P(q_t = s_j \mid q_{t-1} = s_i)$: The probability matrix of state transition; the state transition: $S_i \rightarrow S_j$;

$B = \{ b_j(k) \}, b_j(k) = P(o_t = v_k \mid q_t = s_j)$: The output probability matrix of each state (S_j);

$\pi = \{ \pi_i \}, \pi_i = P(q_1 = s_i), 1 \le i \le N$: The probability vector of the initial state as S_i;

$O = \{ o_1, o_2, ..., o_T \}$: The series of event observation outcomes;

$Q = \{ q_1, q_2, ..., q_T \}$: The series of hidden states.

The calculation of $P(O \mid \lambda)$ is as follows:

$$\begin{aligned} P(O \mid \lambda) &= \sum_{allQ} P(O,\ Q \mid \lambda) \\ &= \sum_{q1,q2,\ldots,qT} \pi_{q1} \cdot b_{q1}(o_1) \cdot a_{q1q2} \cdot b_{q2}(o_2) \cdot a_{q2q3} \cdots a_{qT-1qT} \cdot b_{qT}(o_T) \end{aligned} \tag{11}$$

The calculation of the aforementioned equation is extremely complex and consists of the multiplication of N^T floating-point arithmetic numbers and the addition of N^T-1 floating-point arithmetic numbers. To save calculation time, a forward-backward algorithm was used in this study to calculate $P(O \mid \lambda)$ as follows:

$$\alpha_t(i) = P(o_1,\ o_2,\ ...,\ o_t,\ q_t = s_i \mid \lambda) \tag{12}$$

Recursive start:

$$\alpha_1(i) = \pi_i \cdot b_i(o_1),\ 1 \le i \le N$$

Recursive computation:

$$\alpha_{t+1}(j) = [\sum_{i=1}^{N} \alpha_t(i) \cdot a_{ij}] \cdot b_j(o_{t+1}),\ 1 \le j \le N,\ t = 1,\ 2,\ ...,\ T-1 \tag{13}$$

Recursive end:

$$P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{14}$$

This algorithm requires $(T\text{-}1)N^2 + N$ multiplications and $T \times N$ additions, which substantially reduces the computation complexity. Eq. (14) is the outcome of speech recognition.

4. Experiments

This study used 70 sound records as training files and 25 sound records as recognition files, each containing five phrases. Please see Tables 1 and 2 for more details. Each phrase consisted of three Chinese characters. Each phrase was recorded by four humans. The sampling frequency was 8,000 Hz (mono). The amplitude was 16 bits. The sample size for a frame was

Table 1 Training group.

Sampling frequency: 8 kHz Amplitude: 16 bits Bit rate: 256 kbps	
Speech corpus: Overall, there were five Chinese phrases, each consisting of 14 sounds: how are you, laboratory, air conditioner, motorcycle, fan.	
Training environment	Clean speech sources: four men Recording device: ATH AT-VD4

Table 2 Recognition group.

Sampling frequency: 8 kHz Amplitude: 16 bits Bit rate: 256 kbps	
Speech corpus: Overall, there were five Chinese phrases, each consisting of 5 sounds: how are you, laboratory, air conditioner, motorcycle, fan.	
Training environment	Clean speech sources: four men Recording device: ATH AT-VD4

Table 3 Recognition rate for each phrase.

Phrase (in Chinese)	Frequency (times) of correct recognition	Frequency (times) of incorrect recognition	Recognition rate (%)	Overall recognition rate (%)
How are you	3	2	60%	75%
Laboratory	5	0	100%	
Air conditioner	5	0	100%	
Motorcycle	3	2	60%	
Fan	4	1	80%	

256 bits. The bit rate was 256 kbps. The frame overlapping rate was 50%. The length of the sound file was not fixed. Therefore, endpoint detection was required to remove the blank part of the sound file. The mean energy of the first several frames plus 7.5% of the maximum energy of the frames was set as the threshold. Points that exceeded the threshold were considered as the starting points of the sound. The corresponding equation is as follows:

$$Threshold = 7.5\% \times \max[E(n)] + \frac{1}{K}\sum_{i=1}^{k} E(i);$$

where $E(n)$ denotes the energy of the nth frame, and $\tilde{N}$ is the total number of frames.

The recognition rates for each command (phrase) are revealed in Table 3. It can be seen that the recognition rate for some commands can achieve 100%. And, the overall recognition rate reaches 75%.

5. Conclusion

By applying the HMM in the speech recognition of three Chinese characters, the recognition rate was satisfactory for the 14 samples. A large sample size is not required to apply HMM in the development of numerous functions, which is of interest for commercial applications.

During the experimentation, the recorded sounds for the training and recognition files may have contained sonic bangs occasionally. This phenomenon yielded unsatisfactory endpoint detection outcomes and influenced the correct recognition rates. This problem could be solved by improving the recording hardware quality or removing abnormal portions, which would thus enhance the correct recognition rate. Another problem is that environmental noises always affect the speech command recognition rates. So, the noises cancellation problem is worthwhile to address.

Acknowledgments

This research work was supported by the Ministry of Science and Technology of the Republic of China under contract MOST 108-2221-E-390-018.

References

[1] Buera, L., Miguel, A., Saz, O., Ortega, A., and Lleida, E.

2010. “Unsupervised Data-Driven Feature Vector Normalization With Acoustic Model Adaptation for Robust Speech Recognition.” *IEEE Trans. on Audio, Speech, and Language Processing* 18 (2): 296-309.

[2] Kim, J., and You, B. J. 2011. “Fault Detection in a Microphone Array by Intercorrelation of Features in Voice Activity Detection.” *IEEE Trans. on Industrial Electronics* 58 (6): 2568-71.

[3] Zhan, Y., Kwak, L. H. K. C., and Yoon, H. 2009. “Automated Speaker Recognition for Home Service Robots Using Genetic Algorithm and Dempster-Shafer Fusion Technique.” *IEEE Trans. on Instrumentation and Measurement* 58 (9): 3058-68.

[4] Hsu, C. W., and Lee, L. S. 2009. “Higher Order Cepstral Moment Normalization for Improved Robust Speech Recognition.” *IEEE Trans. on Audio, Speech, and Language Processing* 17 (2): 205-19.

[5] Tsao, Y., and Lee, C. H. 2009. “An Ensemble Speaker and Speaking Environment Modeling Approach to Robust Speech Recognition.” *IEEE Trans. on Audio, Speech, and Language Processing* 17 (5): 1025-37.

[6] Windmann, S., and Haeb-Umbach, R. 2009. “Parameter Estimation of a State-Space Model of Noise for Robust Speech Recognition.” *IEEE Trans. on Audio, Speech, and Language Processing* 17 (8): 1577-90.

[7] Pan, S. T., and Li, X. Y. 2012. “An FPGA-Based Embedded Robust Speech Recognition System Designed by Combining EMD and a Genetic Algorithm.” *IEEE Transactions on Instrumentation & Measurement* 61 (9): 2560-72.

[8] Wang, X. 2004. *Speech Signal Processing*. Kaohsiung: Chuan Hwa Book Co., Ltd.

[9] Lan, M. L. 2006. “Application of Genetic Algorithm to Improve Speech Recognition Based on Artificial Neural Networks.” Master’s thesis, Shu-Te University.

[10] Oppenheim, A. V., Schafer, R. W., and Buck, J. R. 2005. *Discrete-Time Signal Processing*. 2nd ed. London: Pearson.

[11] Blunsom, P. 2004. “Hidden Markov Model.” The University of Melbourne.