Hidden Markov Model for Speech Recognition

Using Modified Forward-Backward Re-estimation Algorithm

Balwant A. Sonkamble[1] and Dr. D. D. Doye[2]

[1] Pune Institute of Computer Technology, Dhankawadi
Pune, Maharashtra State, India

[2] Professor, E&TC Department, SGGSIE&T, Vishnupuri
Nanded, Maharashtra State, India

Abstract

There are various kinds of practical implementation issues for the HMM. The use of scaling factor is the main issue in HMM implementation. The scaling factor is used for obtaining smoothened probabilities. The proposed technique called Modified Forward-Backward Re-estimation algorithm used to recognize speech patterns. The proposed algorithm has shown very good recognition accuracy as compared to the conventional Forward-Backward Re-estimation algorithm.

Keywords: *Forward-Backward Algorithm; Speech Recognition, Parameter Estimation, Viterbi Algorithm, Baum-Welch Algorithm.*

1. Introduction

There was a drastic change in the performance and success of automatic speech recognition systems. The reason is the use of efficient techniques proposed by the researchers time to time. In the early days, the conventional approach uses grammars and templates in the speech recognition system development. The template based approaches are well developed providing the good recognition performance but lack of flexibility was the limitation. The stochastic approaches called Hidden Markov Models (HMM) are the most promising techniques for developing speech recognition systems [1]. HMM is a statistical model generates the model parameters as a collection of values in which the stationary process was approximated by Markov model. The HMMs are very popular due to its simple structure and ease of use. The HMMs key features are its mathematical tractable structure useful to characterize the speech signal very efficiently.

An HMM can be used as a maximum likelihood classifier to compute the probability of a sequence of words given a sequence of acoustic observations. The Forward-Backward recursions were used in HMM as well as computations of marginal smoothing probabilities. The first application of HMMs was speech recognition. The Forward-Backward algorithms are belonging to the general class of algorithms and can be operated on sequence models.

There are various practical implementation issues in HMM including scaling, multiple observation sequences, initial parameter estimates, missing data, and choice of model size and type [1, 4].

In this paper, the main objective is to solve the scaling issue of HMM efficiently using Left-Right model. In the paper organization, Section-II explains brief about the HMM. Section-III, describes about the Modified Forward-Backward Re-estimation algorithm. The experimentation and their results will be discussed in the Section-IV. The conclusion and future work will be explained in Section-V.

2. Hidden Markov Model [1, 2, 3, 5]

The HMM are acted as a more controlled approaches in the recognition/classification based systems. The HMM are the state of art techniques to represent various speech units characteristics within the model parameters. The HMMs are like Markov Chains in which the output symbols and the transitions are probabilistic. The HMMs represent speech as a sequence of observation vectors derived from a probabilistic function of a first-order Markov chain. In speech recognition, the HMMs would output a sequence of *n*-dimensional real-valued vectors. In

each state statistical distribution gives likelihood for each observed vector i.e. for each word will have a different output distribution. The decoding of the speech means computing the most likely word from an unknown utterance.

The HMM are generative models the state space of the hidden variables is discrete, while the observations themselves can either be discrete generated from a categorical distribution or continuous generated from a Gaussian distribution. The HMM has two parameters for modeling. The joint distribution of observations and hidden states means the prior distribution of hidden states called transition probabilities and conditional distribution of observations given states called emission probabilities. In the standard HMM, the algorithm implicitly assumes a uniform prior distribution over the transition probabilities. The transition probabilities are controlled by the hidden states.

The Left-Right model also known as Bakis model, satisfies the property that as time increases the state index increases or stays the same i.e. the states proceed from left to right. It can easily model signals whose properties change over time in a successive manner i.e. speech. The fundamental property of all the left-right HMMs is that the state transition coefficients have the property

$$a_{ij} = 0,\ j < i$$

that is no transitions are allowed to states whose indices are lower than the current state. The initial state probabilities have the property

$$\pi_i = 0, i \neq 1 \text{ otherwise } 1$$

since the state sequence must begin in state 1 and end in state N. In Left-Right model additional constraints are placed on the state transition coefficients in the form of

$$a_{ij} = 0,\ j > i + \Delta$$

where $\Delta = 2$ i.e. no jumps of more than 2 states are allowed. The last state in left-right model is specified by state transition coefficients are

$a_{NN} = 1, a_{Ni} = 0, i < N$.

The constraints are useful to make sure that large changes in state indices do not occur. There is no effect on the re-estimation procedure due to constraints on left-right model because any HMM parameter set to zero throughout the re-estimation procedure.

The goal of HMM parameter estimation is to maximize the likelihood of the data under the given parameter setting [3].

The HMM with discrete probability distributions can be denoted as $\lambda = (A, B, \pi)$. The HMM has three main basic problems, Firstly, the Evaluation problem in which given an HMM λ and a sequence of observations

$$O = o_1, o_2, o_3, ..., o_T,$$

what is the probability of the given observations generated by the model, $p\{O \mid \lambda\}$? This problem is solved using the Forward-Backward Algorithm. Secondly, the decoding problem where given a model λ and a sequence of observations

$$O = o_1, o_2, o_3, ..., o_T,$$

what is the most likely state sequence in the model that produced the observations? This problem is solved using Viterbi algorithm or using the alternative Viterbi algorithm called logarithmic approach. Thirdly, the learning problem also called training where given a model λ and a sequence of observations

$$O = o_1, o_2, o_3, ..., o_T.$$

The model parameters $\lambda = (A, B, \pi)$ are adjusted to maximize $p\{O \mid \lambda\}$. The problem is solved using Forward-Backward Re-estimation algorithm.

3. The Modified Forward-Backward Re-estimation algorithm [7, 8, 9, 11]

The Forward-Backward algorithm acts as an inference algorithm called smoothing based on the principle of dynamic programming which efficiently computes the probability distribution of all hidden state variables given a sequence of observations / emissions. The Forward-Backward algorithm can be used to find the most likely state for any point in time. The Forward-Backward algorithm divided into two stages. In the first stage, the Forward-Backward algorithm computes a set of forward probabilities using auxiliary variable $\alpha_t(i)$ called forward variable. The forward variable defines the probability of the partial observation sequence $o_1, o_2, o_3, ..., o_T$ has low complexity. It terminates at the state i and the mathematical representation is in the form of,

$$\alpha_t(i) = p\{o_1, o_2, \ldots, o_t, q_t = i \mid \lambda\} \cdots (1)$$

The recursive function can be written as

$$\alpha_{t+1}(j) = b_j(o_{t+1}) \sum_{i=1}^{N} \alpha_t(i) a_{ij}, 1 \le j \le N, 1 \le t \le T-1 \cdots (2)$$

where, $\alpha_1(j) = \pi_j b_j(o_1), 1 \le j \le N$

where $\alpha_T(i), 1 \le i \le N$

can be calculated and the required probability is given by,

$$p\{O \mid \lambda\} = \sum_{i=1}^{N} \alpha_T(i) \cdots (3)$$

In the second stage, the backward variable $\beta_t(i)$ is used as the probability of the partial observation sequence $o_{t+1}, o_{t+2}, o_{t+3,}..., o_T$ which is the probability of the partial observation sequence from time $t+1$ to the end, given state i at time t and model λ. It is mathematically represented as,

$$\beta_t(i) = p\{o_{t+1}, o_{t+2}, \ldots, o_T \mid q_t = i, \lambda\} \cdots (4)$$

and for calculating the $\beta_t(i)$ effectively, the representation can be given by,

$$\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j) a_{ij} b_j(o_{t+1}), 1 \le i \le N, 1 \le t \le T-1 \cdots (5)$$

where, $\beta_T(i) = 1, 1 \le i \le N$ and the required probability is given by,

$$p\{O \mid \lambda\} = \sum_{i=1}^{N} \pi_i b_i(o_1) \beta_1(i) \cdots (6)$$

Where

$$\alpha_t(i)\beta_{t(i)} = p\{O, q_t = i \mid \lambda\}, 1 \le i \le N, 1 \le t \le T \cdots (7)$$

The following representation used to calculate $p\{O \mid \lambda\}$, by using both forward and backward variables

$$p\{O \mid \lambda\} = \sum_{i=1}^{N} p\{O, q_t = i \mid \lambda\} = \sum_{i=1}^{N} \alpha_t(i) \beta_t(i) \cdots (8)$$

The final probability is computed using the smoothed probability values of the forward and backward probabilities.

The smoothed probability values must be scaled and its entries sum to 1. These are obtained by applying scaling factor. In the standard HMM implementation no need to use scaling factor in the logarithmic approach. We are proposing to apply scale to the backward probabilities. The backward probability vectors actually represent the likelihood of each state at time t given the future observations. These vectors are proportional to the actual backward probabilities and the result has to be scaled an additional time. As per many researchers, the forward are sufficient to calculate the most likely final state but omitting the scaling factor applied on the backward probabilities decreases the performance. In the proposed experiment the speech recognition accuracy was increased by adding scaling factor to the backward probabilities. These backward probabilities are combined with the initial state vector to provide the most probable initial state for given the observations. The forward and backward probabilities need only be combined to infer the most probable states between the initial and final points.

The scaling factor can be expressed as

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

applied with both the forward and backward probabilities.

By adding scaling factor the expressions becomes

$$\alpha_t(i) = c_t \alpha_t(i) \text{ and } \beta_t(i) = c_t \beta_t(i).$$

The Baum-Welch algorithms key feature was to provide guaranteed convergence. The Baum-Welch algorithm uses the optimizing criteria based on the maximum likelihood (ML). The ML criteria maximizes the probability of a given sequence of observations O^w, belonging to a given class w, given the HMM λ_w of the class w, with respect to the parameters of the model λ_w. This probability is the total likelihood of the observations for all classes or words and can be expressed mathematically as

$$L = p\{O^w \mid \lambda_w\} \cdots (9)$$

The ML criterion for one class or word can be written as,

$$L = p\{O \mid \lambda\} \cdots (10)$$

However there is no known way to analytically solve for the model $\lambda = (A, B, \pi)$, which maximize the quantity L. The model parameters are chosen which are locally maximized. The iterative procedure applied derived from simple occurrence counting arguments or through calculus to maximize the auxiliary quantity

$$Q(\lambda, \bar{\lambda}) = \sum_{q} p\{q \mid O, \lambda\} \log[p\{O, q, \bar{\lambda}\}] \cdots (11)$$

Where $\bar{\lambda}$ = []. The Baum-Welch algorithm also known as Forward-Backward Re-estimation algorithm can be described by defining two more auxiliary variables, in addition to the forward and backward variables. These variables can be expressed in terms of the forward and backward variables.

The first variable is defined as the probability of being in state i at $t=t$ and in state j at $t=t+1$. This can be formally represented as,

$$\xi_t(i,j) = p\{q_t = i, q_{t+1} = j \mid O, \lambda\} \cdots (12)$$

The above expression can be rewritten as,

$$\xi_t(i,j) = \frac{p\{q_t = i, q_{t+1} = j, O \mid \lambda\}}{p\{O \mid \lambda\}} \cdots (13)$$

Using forward and backward variables this can be expressed

as,

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} \beta_{t+1}(j) b_j(o_{t+1})}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i) a_{ij} \beta_{t+1}(j) b_j(o_{t+1})} \cdots (14)$$

the second variable is the a posteriori probability,

$$\gamma_t(i) = p\{q_t = i \mid O, \lambda\} \cdots (15)$$

that is the probability of being in state i at $t=t$, given the observation sequence and the model. In forward and backward variables this can be expressed by,

$$\gamma_t(i) = \left[\frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)} \right] \cdots (16)$$

The relationship between $\gamma_t(i)$ and $\xi_t(i,j)$ is given by,

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j), 1 \le i \le N, 1 \le t \le M \cdots (17)$$

The parameters of the HMM are updated to maximize the quantity of $p\{O \mid \lambda\}$ in the Baum-Welch learning process. The starting model $\lambda = (A, B, \pi)$, calculates the $'\alpha's$ ‘, $'\beta's'$ and then $'\xi's$, $'\gamma's$. The HMM parameters are updated using the following steps known as re-estimation formulas.

$$\pi_i = \gamma_1(i), 1 \le i \le N \cdots (18)$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, 1 \le i \le N, 1 \le j \le N \cdots (19)$$

$$\bar{b}_j(k) = \frac{\sum_{\substack{i=1 \\ o_t = v_k}}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}, 1 \le j \le N, 1 \le k \le M \cdots (20)$$

The preprocessing part of the system gives out a sequence of observation vectors

$$O = \{o_1, o_2, \ldots, o_N\} \cdots (21)$$

The initial observation vector parameter values for each of the HMMs are

$$\lambda_i, 1 \le i \le N \cdots (22)$$

Lastly, the likelihoods can be calculated using the forward and backward variables through the following equation

$$L_{likelihood} = \sum_{i \in \lambda_i} \alpha_t(i)\beta_t(i) = \alpha_T(i) \cdots (23)$$

4. The Experimentation and Results

The TI46 database [12] is used for experimentation. There are 16 speakers from them 8 male speakers and 8 female speakers. The numbers of replications are 26 for utterance by each person. The total database size is 4160 utterances of which1600 samples used for training and remaining samples are used for testing of 10 words that are numbers in English 1 to 9 and 0 are sampled at a rate of 8000 Hz. A vector of 12 Linear Predicting Coding Cepstrum coefficients was obtained and provided as an input to vector quantization to find codewords for each class. The VQ codebook [9, 10] maps each continuous observation vector into a discrete codebook index. The vector quantized values are provided to Left-Right model for modeling. In the recognition phase likelihoods will be calculated for matching and recognizing the digits.

Table-1 shows the results obtained from Left-Right model using 3-states from which we can see that the average accuracy is increased by 4% as compared to the proposed algorithm.

Table-1 Accuracy in % for Left-Right model using 3-states.

digits	LRfb3	LRmfb3
0	80	**100**
1	100	**100**
2	100	**100**
3	100	**100**
4	100	**93.33**
5	100	**100**
6	86.67	**100**
7	100	**100**
8	93.33	**93.33**
9	86.67	**80**
Avg	92.67	**96.67**

Table-2 shows the results obtained from Left-Right model using 5-states from which we can see that the average accuracy is more compared to the proposed algorithm the reason is that the utterances considered in isolation.

Table-2 Accuracy in % for Left-Right model using 5-states.

digits	LRfb5	LRmfb5
0	93.33	100
1	100	100
2	100	80
3	100	100
4	93.33	93.33
5	93.33	100
6	100	93.33
7	93.33	86.67
8	93.33	80
9	86.67	73.33
Avg	95.33	90.66

Fig. 3 shows the recognition accuracy using Modified Forward-Backward Re-estimation algorithm compared with the conventional Forward-Backward Re-estimation algorithm with 3-states.

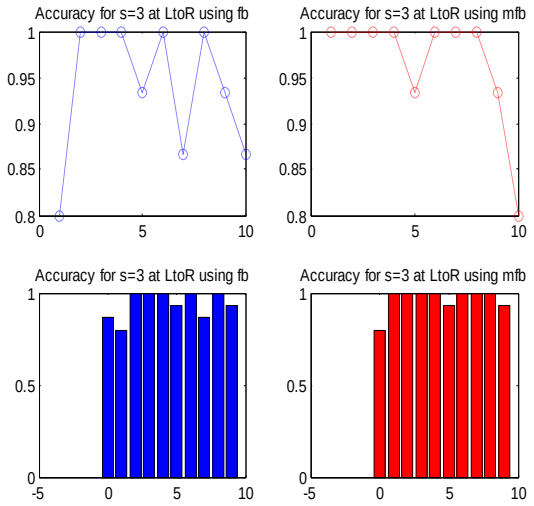

Fig. 3 The Recognition Accuracy for 3-state Left-Right Model.

Fig. 4 shows the recognition accuracy using Modified Forward-Backward Re-estimation algorithm compared with the conventional Forward-Backward Re-estimation algorithm with 5-states.

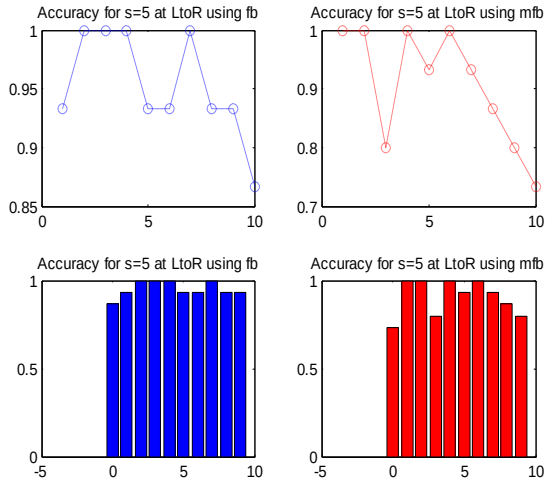

Fig. 4 The Recognition Accuracy for 5-state Left-Right Model.

4. Conclusions

The proposed Modified Forward-Backward Re-estimation algorithm has shown improved recognition accuracy as compared to conventional Forward-Backward Re-estimation algorithm using Left-Right model. The recognition accuracy is increased by 4% in the 3-state model whereas the recognition accuracy of 5-state model obtained with the conventional algorithm is more because Left-Right model used was single observation sequence. The experimentation is done for isolated words which requires minimum number of states and hence the recognition accuracy for 5-state model constant. The states above 5-states can be used in continuous speech recognition systems.

References

[1] Lawrence R. Rabiner, "A Tutorial on Hidden Markov Model and Selected Application in Speech Recognition", Proceedings of the IEEE, Vol. 77, No. 2, pp. 257 – 286, 1989.

[2] Qystein Birkenes,Tomoko Matsui, Kunio Tanabe, Sabato Marco Siniscalchi, Tor Andre Myrvoll, and Magne Hallstein Johnsen, "Penalized Logistic Regression with HMM Log-Likelihood Regressors for Speech Recognition", IEEE Transactions on Audio, Speech, and Language Processing Vol. 18, No. 6, pp. 1440-1454, August 2010.

[3] Xiong Xiao, Jinyu Li, Eng Siong Chng, Haizhou Li, Chin-Hui Lee, "A Study on Hidden Markov Models Generalization Capability for Speech Recognition", Proceedings of ASRU-2009, pp. 255-260, 2009.

[4] Shun-Zheng Yu and Hisashi Kobayashi, "Practical Implementation of an Efficient Forward–Backward Algorithm for an Explicit-Duration Hidden Markov Model", IEEE Transactions on Signal Processing, Vol. 54, no. 5, pp. 1947-1955, May 2006.

[5] Mark Gales and Steve Young, "The Application of Hidden Markov Models in Speech Recognition", Journal of Foundations and Trends in Signal Processing, Vol. 1, No. 3, pp. 195–304, 2007.

[6] Levinson S., Rabiner L., Sondhi M., "Speaker independent isolated digit recognition using hidden Markov models", Proceedings of ICASSP '83, Vol. 8, pp. 1049–1052, 1983.

[7] Juang B., Rabiner L., Levinson S., Sondhi M., "Recent developments in the application of hidden Markov models to speaker-independent isolated word recognition", Proceedings of ICASSP '85, Vol. 10, pp. 9–12, 1985.

[8] Sugawara K., Nishimura M., Toshioka K., Okochi M., Kaneko T., "Isolated word recognition using hidden Markov models", Proceedings of ICASSP '85, Vol. 10, pp.1-4, 1985.

[9] Cheung Y., Leung, S., "Speaker-independent Isolated Word Recognition using word-based vector quantization and hidden Markov models", Proceedings of ICASSP' 87, Vol. 12, pp. 1135–1138, 1987.

[10] Falkhausen M., Euler S.A., Wolf D., "Improved training and recognition algorithms with VQ-based hidden Markov models", Proceedings of ICASSP-90, Vol. 1, pp. 549–552, 1990.

[11] Peinado, A.M.; Lopez, J.M.; Sanchez, V.E.; Segura, J.C.; Rubio Ayuso, A.J., "Improvements in HMM-based isolated word recognition system", IEE Proceedings of Communications, Speech and Vision, Vol. 138 , No. 3, pp. 201–206, 1991.

[12] NIST, "TI46 CD", September, 1991.

Balwant A. Sonkamble received his BE (Computer science and Engineering in 1994 and M. E. (Computer Engineering) in 2004. Currently he is research scholar at SGGS College of Engineering and Technology, Vishnupuri, Nanded (MS)–INDIA. He is working as a Associate Professor in Computer Engineering at Pune Institute of Computer Technology, Pune, India. His research areas are Speech Recognition and Artificial Intelligence.

D D Doye received his BE (Electronics) degree in 1988, ME (Electronics) degree in 1993 and Ph. D. in 2003 from SGGS College of Engineering and Technology, Vishnupuri, Nanded (MS) – INDIA. Presently, he is working as Professor in department of Electronics and Telecommunication Engineering, SGGS Institute of Engineering and Technology, Vishnupuri, Nanded. His research fields are speech processing, fuzzy neural networks and image processing.