Forward-Backwards Training of Hybrid HMM/BN Acoustic Models

Konstantin Markov[1,2]*, Satoshi Nakamura*[1,2]

[1]National Institute of Information and Communications Technology, Japan
[2]ATR Spoken Language Communication Research Lab., Japan
{konstantin.markov,satoshi.nakamura}@atr.jp

Abstract

In this paper, we describe an application of the Forward-Backwards (F-B) algorithm for maximum likelihood training of hybrid HMM/Bayesian Network (BN) acoustic models. Previously, HMM/BN parameter estimation was based on a Viterbi training algorithm that requires two passes over the training data: one for BN learning and one for updating HMM transition probabilities. In this work, we first analyze the F-B training for a conventional HMM and show that the state PDF parameter estimation is analogous to weighted-data classifier training. The gamma variable of the Forward-Backwards algorithm plays the role of the data weight. From this perspective, it is straightforward to apply F-B-based training to the HMM/BN models since the BN learning algorithm allows training with weighted data. Experiments on accented speech (American, British and Australian English) show that F-B training outperforms the previous Viterbi learning approach and that the HMM/BN model achieved better performance than the conventional HMM.

Index Terms: forward-backwards algorithm, HMM/BN, weighted data training, accent modeling.

1. Introduction

In recent years, research activities in speech modeling frameworks other than HMM have intensified. Dynamic Bayesian Networks (DBN) [1] have been successfully applied in the automatic speech recognition (ASR) [2, 3]. Also, our group at ATR has proposed an HMM/BN model [4] that proved itself to be effective replacement of the HMM [5].The HMM/BN model can be viewed as both a generalization of HMM and as a DBN with temporal topology constraints. What distinguishes HMM/BN from HMM is the great flexibility of state probability modeling that BN offers. On the other hand, training and implementation of the HMM/BN is much easier and more tractable than DBN and is very similar to HMM.

Structurally, the HMM/BN model is analogous to the hybrid HMM/Neural Network model (NN). The difference is that instead of an NN, the HMM is coupled with a BN. Initially, for the HMM/BN training, we used the same approach as for the HMM/NN - the Viterbi paradigm. It consists of three alternating steps: Viterbi alignment, BN training and HMM probabilities update. HMMs can also be trained using the Viterbi algorithm, but it is known that forward-backwards training yields better models. Here, we discuss how to apply the F-B algorithm for HMM/BN training. First, we analyze the ML parameter estimates for the standard HMM and show that for each state the F-B training is equivalent to learning with weighted data where the *gamma* variable plays the role of the weight. Such weighted-data training is used in the boosting algorithms that have also been applied for improving HMM- as well as DBN-based systems [6, 7]. Next, we show that since BN learning can be done with weighted data, it is straightforward to apply an F-B algorithm to HMM/BN training.

2. Hybrid HMM/BN Model

The HMM/BN model is a combination of an HMM and a Bayesian Network. Temporal speech characteristics are modeled by the HMM state transitions while the HMM states' probability distributions are represented by the BN. A block diagram of the HMM/BN is shown in Fig.1.

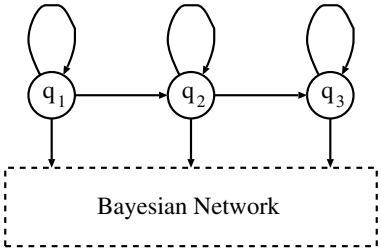

Figure 1: HMM/BN model structure.

By definition, a Bayesian Network represents a joint probability distribution of a set of random variables $Z_1, Z_2 \dots Z_N$ and is expressed by a directed acyclic graph (DAG), where each node corresponds to a unique variable. Arcs between the nodes show the conditional dependencies of the BN variables. Immediate predecessors of variable Z_i are called its *parents* and are referred to as $Pa(Z_i)$. The BN joint probability distribution function can be factored as [8]

$$P(Z_1, Z_2 \dots Z_N) = \prod_{i=1}^{N} P(Z_i|Pa(Z_i)). \quad (1)$$

When all BN variables are observable, i.e., in the fully observable case, the maximum likelihood (ML) approach to parameter estimation can be easily applied. In this case, given the training data $o_1, o_2 \dots o_T$, the log-likelihood function is [3]

$$\begin{aligned} L &= \log \prod_{t=1}^{T} Pr(o_t|G) \\ &= \sum_{i=1}^{N} \sum_{t=1}^{T} \log(Z_i|Pa(Z_i), o_t) \end{aligned} \quad (2)$$

where G denotes the BN. We can see that this function decomposes into a series of terms, one per variable. Therefore, the ML training is essentially a parameter estimation of each node's conditional probability density (CPD) given its local data $\{o_t(Z_i, Pa(Z_i))\}$.

The BN of the HMM/BN model consists of at least two variables: state variable Q and observation (speech) variable X connected with a single arc from Q to X. In this case, the HMM/BN model is equivalent to the conventional HMM. The advantage of using BN as state probability model is that it is easy to add other variables representing different speech features or variability factors. The easiest HMM/BN implementation into an ASR system is when all variables except X and Q are discrete and assumed hidden (for training, however, they can be observable). Then, for an arbitrary BN having joint pdf $P(X, Q, Z_1 \dots Z_N)$, we have the state output probability

$$P(X|Q) = \sum_{z_1} \cdots \sum_{z_N} \prod_{i=1}^{N} P(Z_i = z_i|Pa(Z_i))P(X|Pa(X)) \quad (3)$$

which actually represents a Gaussian mixture where the product terms $\prod_{i=1}^{N} P(Z_i = z_i|Pa(Z_i))$ are the mixture weights and $P(X|Pa(X))$ are the Gaussian functions.

3. Forward-Backwards Algorithm

The basic idea of the F-B algorithm is to recursively compute two variables [9]. The first one, called *alpha*, is obtained in the forward pass and is the probability of observing input sequence $x_{1:t}$ and state q_i at time t, i.e. $\alpha_t(i) = P(x_{1:t}, q_i^t)$. The second variable, *beta*, is obtained in the backwards pass and is the probability of observing input sequence $x_{t+1:T}$ given state q_i at time t, $\beta_t(i) = P(x_{t+1:T}|q_i^t)$. Then, a third variable, *gamma*, is defined as $\gamma_t(i) = P(q_i^t|x_{1:T})$ and it is the probability of being in state q_i at time t given the entire input sequence $x_{1:T}$. This variable can be expressed in terms of forward and backwards variables as

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_j \alpha_t(j)\beta_t(j)} \quad (4)$$

and it has the following property: $\sum_i \gamma_t(i) = 1$.

Often, HMM parameters are learned by using the Viterbi algorithm which can be viewed as a special case of F-B where $\gamma_t(i)$ is equal to either zero or one.

3.1. F-B-based HMM parameter estimation

The maximum likelihood estimation of the HMM parameters uses the F-B algorithm's gamma variable, and the procedure is well known. Here, we provide only the final equations for the state PDF parameters.

For discrete HMM, where state PDF is represented by a probability table, the estimate of probability of observing symbol $V_k, k = 1, 2 \dots K$ for state q_i is

$$\hat{P}_i(V_k) = \frac{\sum_{t=1}^{T} \gamma_t(i)\delta(x_t = V_k)}{\sum_{t=1}^{T} \gamma_t(i)} \quad (5)$$

where $\delta()$ is one when $x_t = V_k$ and zero otherwise.

When HMM state PDF is modeled by a mixture of Gaussian functions $\{c_m, N(; \mu_m, \Sigma_m)\}, m = 1, 2 \dots M$, then the estimate for the mixture weights of state q_i is

$$\hat{c}_{m_i} = \frac{\sum_{t=1}^{T} \gamma_t(i)p(m_i|x_t)}{\sum_{t=1}^{T} \gamma_t(i)} \quad (6)$$

and for the means we have:

$$\hat{\mu}_{m_i} = \frac{\sum_{t=1}^{T} \gamma_t(i)p(m_i|x_t)x_t}{\sum_{t=1}^{T} \gamma_t(i)p(m_i|x_t)} \quad (7)$$

and the estimates of the covariance matrices are

$$\hat{\Sigma}_{m_i} = \frac{\sum_{t=1}^{T} \gamma_t(i)p(m_i|x_t)(x_t - \mu_{m_i})(x_t - \mu_{m_i})'}{\sum_{t=1}^{T} \gamma_t(i)p(m_i|x_t)} \quad (8)$$

where $p(m_i|x_t)$ is the posterior of the mixture component m_i and is calculated as:

$$p(m_i|x_t) = \frac{c_{m_i} N(x_t; \mu_{m_i} \Sigma_{m_i})}{\sum_{k=1}^{M} c_{k_i} N(x_t; \mu_{k_i} \Sigma_{k_i})} \quad (9)$$

3.2. F-B based HMM/BN parameter estimation

Analyzing the HMM parameter estimation equations from the previous section, we can see that the gamma variable depends on the time index t and state ID but does not depend on the functional form of the state PDF. Therefore, we can associate each input data x_t with a set of gammas, one for each state i, $\{\gamma_t(i), i = 1, \dots, S\}$. Next, let's assume that each state represents a different classifier. Accordingly, $\gamma_t(i)$ can be interpreted as a weight that shows how important is for classifier i to correctly classify input data x_t. Such interpretation makes state parameter learning analogous to weighed-data classifier training which is one of the main concepts of boosting algorithms, particularly the AdaBoost algorithm [10]. An essential point in this case is that boosted learning can be applied to any type of classifier as long as its training algorithm allows training with weighted data.

As described in Section 2, in the fully observable case, the BN training can be decomposed to ML estimation of each node's parameters. If the F-B algorithm is applied, this becomes an estimation using weighted data. Since the weights (gammas) are different for each state ID, for each node j, the PDF learning is done with the node's local data conditioned by the state variable, i.e. $\{\gamma_t(i), o_t(Z_j, Pa(Z_j)|Q = i)\}$. Let's consider, for example, the estimation of discrete node CPD, which is represented by a probability table defined by $\{\theta_{jkli} = P(Z_j = k|Pa(Z_j) = l, Q = i)\}$. Consequently, the ML estimate from weighted data is

$$\hat{\theta}_{jkli} = \frac{\sum_{t=1}^{T} \gamma_t(i)\delta(Z_j = k, Pa(Z_j) = l, Q = i)}{\sum_{t=1}^{T} \gamma_t(i)} \quad (10)$$

and it is analogous to Eq.(5). For continuous nodes whose CPD is represented by a Gaussian function or Gaussian mixture, parameter estimates are essentially the same as Eqs.(6-8) except that instead of x_t, all summations are done over the node's local data conditioned on the state ID.

Estimation of the HMM/BN transition probabilities is the same as for the conventional HMM.

4. Experiments

In our experiments, we used a database of accented English read speech consisting of utterances from American (US), British (BRT) and Australian (AUS) speakers. There are 50 male and 50 female speakers per accent (300 in total) and about 300 utterances per speaker. The text material includes phonetically balanced sentences from the TIMIT database and transcripts of travel related conversations. From each accent group 90 speakers were selected for training (270 in total). For evaluation, only 100 travel related utterances from each of the remaining 30 speakers (3000 utterances in total) were used. Speech data were preprocessed in a standard way and 25-dimensional feature vectors (12MFCC+12ΔMFCC+ΔE) were obtained with 20ms windows at a 10ms rate.

Both the speaker's accent and his/her gender are speech variability factors that can be easily modeled by the BN. A BN topology that reflects dependencies between speech X, accent A, gender G and the HMM state Q is shown in Fig. 2.

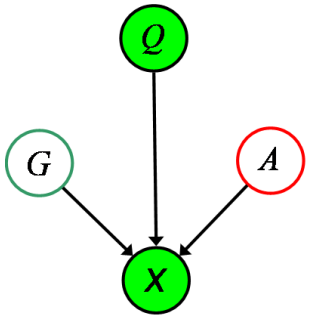

Figure 2: BN topology used in the experiment. Variable G represents speaker's gender and variable A is speaker's accent.

Since for each utterance the speaker's accent and gender are known, the BN is fully observable during the training. During recognition however, these factors are unknown and thus variables G and A are considered hidden. In this case, according to Eq.(3), the state output probability is calculated as follows:

$$P(x_t|q_i) = \sum_{a_k} \sum_{g_l} P(a_k)P(g_l)P(x_t|a_k, g_l, q_i) \qquad (11)$$

where $a_k = \{US, AUS, BRT\}$ and $g_l = \{M, F\}$. The $P(a_k)$ and $P(g_l)$ are accent and gender prior probabilities respectively. The conditional probability $P(x_t|a_k, g_l, q_i)$ is represented by a mixture of Gaussian functions.

First, using all of the training data we trained a standard triphone HMM using the MDL-SSS state splitting algorithm [11] which resulted in an acoustic model with 3275 shared states. This model, denoted as HMM1, served both as a baseline and as a bootstrap model for the HMM/BN. Next, from the male and female parts of the data we built two gender dependent models, denoted as HMM2, by retraining the HMM1. In this way, all models have the same state structure. Similarly, three accent dependent acoustic models, HMM3, were trained using the US, British and Australian speakers' data separately. Initially, all models had 6 component Gaussian mixtures per state (3 for each gender dependent model and 2 for each accent dependent model, but 6 in total), and subsequently the mixture size was increased to 18, 30 and 42.

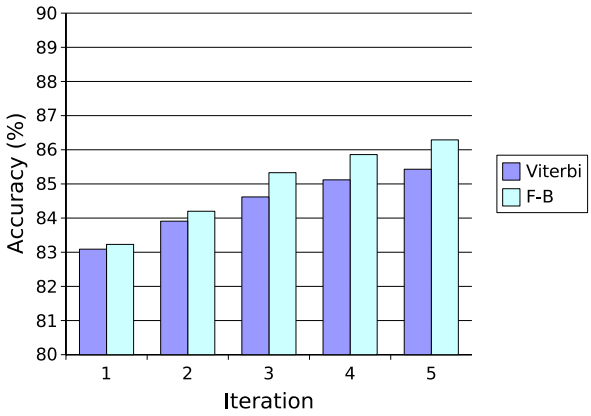

Figure 3: Recognition rates of HMM/BN (18 mix./state) trained by F-B and Viterbi algorithms.

The HMM/BN model was initialized using the HMM1 which means that they have the same state number and the same state tying structure. In order to match the mixture size of the baseline model, each of the six $P(x_t|a_k, g_l, q_i)$ per state was modeled by 1, 3, 5 and 7 Gaussian component mixtures. Since the amount of training data for each condition (gender, accent) is roughly the same, the estimated accent and gender prior probabilities were $P(a_k) \approx 0.33$ and $P(g_l) \approx 0.5$ respectively. The HMM/BN parameters were estimated as described in Section 3.2. Five iterations of F-B training were done for all of the models.

Evaluation experiments were performed using standard bi-gram and tri-gram language models (LM) for the decoding and word lattice rescoring passes, respectively. They were trained on about 600,000 travel related sentences. The test data perplexity is 27.6 for the bi-gram and 19.2 for the tri-gram. Vocabulary consists of about 35,000 words and there are no out-of-vocabulary words. The lexicon is based on standard American English word pronunciation with an average 1.2 pronunciation variants per word. Decoder parameters, such as LM scale and beam width were kept constant in all evaluations.

First, we compared the F-B and Viterbi algorithms and for this purpose, one HMM/BN model with a total of 18 Gaussians per state was learned using Viterbi training. Word recognition rates obtained in this experiment are shown in Fig. 3.

Next, we compared the HMM/BN model with the baseline and the gender and accent dependent HMM models. The recognition with multiple models, i.e. with HMM2 and HMM3, was performed by parallel decoding, and the final hypothesis was chosen as that with the highest score among each model's 1-best hypothesis. Word recognition accuracies for the entire test set are presented in Fig. 4.

The best result was obtained from the HMM/BN, but it outperformed the HMM only when the mixture size is large. This can be explained by the fact that in addition to speaker accent and gender, there are many other speech variability factors that to some extent are learned implicitly even by a small-size mixtures, as in the case of HMM. In the HMM/BN case, however, mixtures are conditioned on both the gender and accent, and when the total number of mixture components is small, there are too few Gaussians per condition resulting in a poor modeling of the other speech variabil-

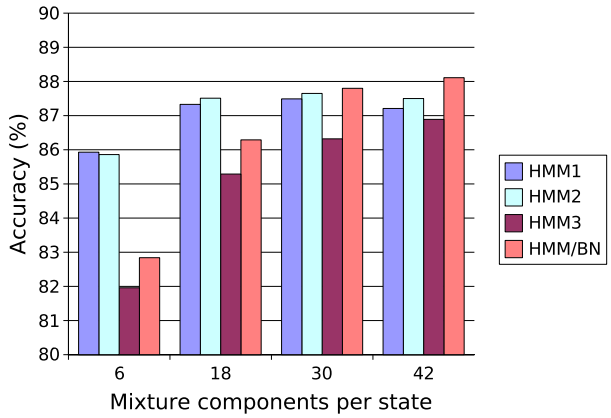

Figure 4: Total recognition rates for HMM and HMM/BN.

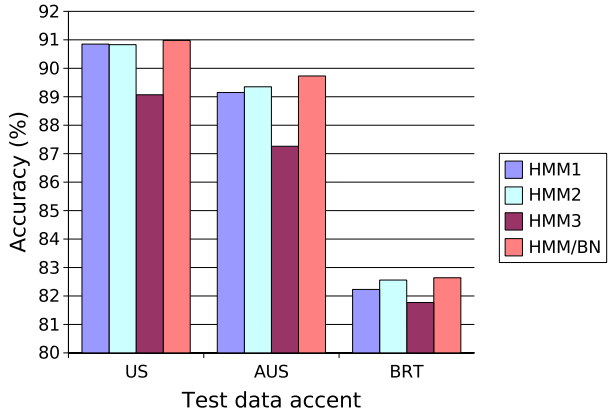

Figure 5: HMM and HMM/BN (42 mix./state) recognition rates for different English accents.

ities.

Recognition rates for each English accent are presented in Fig. 5 which shows that both the HMM/BN and HMM models give better performance for the American accent than for the British or Australian accents. This is, probably, due to the pronunciation lexicon which includes only standard American English word pronunciations.

5. Conclusions

In this study, we showed how to apply a Forward-Backward algorithm for maximum likelihood estimation of the HMM/BN model parameters. We experimented with fully observable BN during training, but even in the partially observable case, the F-B training procedure remains the same.

We compared HMM/BN models trained using both F-B and Viterbi algorithms, and, as expected, F-B training yielded better-performing models. The experiments also showed, that the HMM/BN can outperform a conventional HMM having the same state structure and number of parameters.

6. References

[1] T. Dean and K. Kanazawa, "Probabilistic temporal reasoning," in *AAAI*, 1988, pp. 524–528.

[2] G. Zweig and S. Russell, "Probabilistic modeling with Bayesian Networks for automatic speech recognition," in *Proc. ICSLP*, 1998, pp. 3010–3013.

[3] K. Murphy, *Dynamic Bayesian Networks: Representation, Inference and Learning*, Ph.D. thesis, University of California, Berkeley, 2002.

[4] K. Markov and S. Nakamura, "A hybrid HMM/BN acoustic model for automatic speech recognition," *IEICE Trans. Inf. & Syst.*, vol. E86-D, no. 3, pp. 438–445, 2003.

[5] K. Markov, S. Nakamura, and J. Dang, "Integration of articulatory dynamic parameters in HMM/BN based speech recognition system," in *Proc. ICSLP*, 2004, pp. 774–777.

[6] C. Meyer, "Utterance-level boosting of HMM speech recognizers," in *Proc. ICASSP*, 2002, vol. 1, pp. 109–112.

[7] V. Garg, A.; Pavlovic and J.M. Rehg, "Boosted learning in Dynamic Bayesian Networks for multimodal speaker detection," *Proceedings of the IEEE*, vol. 91, no. 9, pp. 1355 – 1369, Sept. 2003.

[8] F. Jensen, *An introduction to Bayesian networks*, UCL Press, 1998.

[9] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–285, Feb. 1989.

[10] Y. Freund and R. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[11] T. Jitsuhiro, T. Matsui, and S. Nakamura, "Automatic generation of non-uniform HMM topologies based on the MDL criterion," *IEICE Trans. Inf. & Syst.*, vol. E87-D, no. 8, 2004.