PAPER Special Section on Statistical Modeling for Speech Processing

Using Hybrid HMM/BN Acoustic Models: Design and Implementation Issues

Konstantin MARKOV[†a)], ***Nonmember and*** **Satoshi NAKAMURA**[†], ***Member***

SUMMARY In recent years, the number of studies investigating new directions in speech modeling that goes beyond the conventional HMM has increased considerably. One promising approach is to use Bayesian Networks (BN) as speech models. Full recognition systems based on Dynamic BN as well as acoustic models using BN have been proposed lately. Our group at ATR has been developing a hybrid HMM/BN model, which is an HMM where the state probability distribution is modeled by a BN, instead of commonly used mixtures of Gaussian functions. In this paper, we describe how to use the hybrid HMM/BN acoustic models, especially emphasizing some design and implementation issues. The most essential part of HMM/BN model building is the choice of the state BN topology. As it is manually chosen, there are some factors that should be considered in this process. They include, but are not limited to, the type of data, the task and the available additional information. When context-dependent models are used, the state-level structure can be obtained by traditional methods. The HMM/BN parameter learning is based on the Viterbi training paradigm and consists of two alternating steps - BN training and HMM transition updates. For recognition, in some cases, BN inference is computationally equivalent to a mixture of Gaussians, which allows HMM/BN model to be used in existing decoders without any modification. We present two examples of HMM/BN model applications in speech recognition systems. Evaluations under various conditions and for different tasks showed that the HMM/BN model gives consistently better performance than the conventional HMM.

key words: *HMM/BN, acoustic model, Bayesian network*

1. Introduction

For many years, since the introduction of the HMM for speech recognition [1], [2], it has been the dominating tool for acoustic modeling of speech signals. A lot of research has been directed into improving and extending the HMM framework and significant advances have been achieved. However, the pace of performance improvement has significantly slowed down lately, suggesting that to some extent we may soon reach or may have already reached the limit of HMM modeling power. As a consequence, the number of studies pursuing new, beyond-HMM approaches has increased recently.

One promising research direction is the application of Bayesian Networks (BN) as speech models. Bayesian Networks (BN) have attracted researchers' attention because they can model complex joint probability distributions of many different (discrete and/or continuous) random variables in well structured and easy-to-represent ways. Especially suitable for modeling temporal speech characteristics is the Dynamic BN (DBN) [3]–[5]. DBN is regarded as a generalization of the HMM, which in addition to speech spectral information can easily incorporate additional knowledge, such as the sub-band correlation [6], speaking rate [7], etc. In [8], acoustic features are easily supplemented with pitch information within the framework of DBN. A combination of features coming from different modalities, i.e. audio and video, can be easily implemented with the help of DBN [9]. Another advantage of Bayesian Networks is that additional features that are difficult to estimate reliably during recognition may be left hidden, i.e. unobservable.

Our research group at ATR Spoken Language Communication Research Labs has been developing a new acoustic model called a hybrid HMM/BN. In the HMM/BN, temporal characteristics of speech signal are modeled by HMM state transitions and a BN is used to model HMM state distributions. The advantage of this is that the existing methods for HMM design, training and recognition can be used without significant modifications since the HMM/BN behaves essentially as a conventional HMM. We have successfully applied this model in several tasks such as noisy speech recognition [10], large vocabulary [11] or phoneme recognition [12]. In all these cases, different additional information such as noise type and SNR value, speaker gender and pitch frequency or articulatory parameters has been integrated in the state probability density functions by means of BN. By incorporating non-spectrum-based information in our model we were able to increase its performance, which was consistently better than that of the conventional HMM. The HMM/BN model can be regarded as an extension of the HMM or as a DBN with a constraint on the network topology; i.e., temporal dependencies are allowed only between state variables. What makes the HMM/BN different from the traditional HMM is the efficient and flexible modeling of the state probability density by the BN. On the other hand, the hybrid model does not require the complicated and computationally expensive inference algorithms that are used with the DBN.

In this paper, based on our experience with the HMM/BN, we describe and discuss several practical issues that may arise when developing and implementing this model. The first things to be decided in the model design are the state structure and the BN topology. Since at the state level the HMM/BN is equivalent to the traditional HMM, we can use the same methods to develop the state structure. For small tasks and context-independent unit models, this

Manuscript received July 11, 2005.
Manuscript revised October 5, 2005.

[†]The authors are with the ATR Spoken Language Communication Research Labs., Kyoto-fu, 619–0288 Japan.
a) E-mail: konstantin.markov@atr.jp

DOI: 10.1093/ietisy/e89–d.3.981

includes only setting a proper number of states. Furthermore, the structure is almost always left-to-right. When a context-dependent (for example: triphone) unit model is required, traditional state clustering and tying techniques can be applied. Once the state structure is fixed, the BN topology has to be chosen. It is in this step we can integrate different speech information sources. Dependencies can be set according to prior knowledge or data correlation analysis. In this way, we can impose a knowledge-based structure on the state output pdf and achieve a more precise model.

The HMM/BN model training involves two main steps: estimation of state transition probabilities and training of the BN itself. This procedure is based on the Viterbi training algorithm, where the two steps are performed sequentially at each iteration and the parameters learned at the previous step are fixed. In general, Bayesian statistical methods are used for BN learning [13]. The most simple algorithms apply to tree-structured BN with only discrete variables [14], [15]. For non-tree structures, the "junction tree" (or JLO) algorithm [16], [17] is widely used. When the complexity of the BN makes these algorithms infeasible, a number of approximate algorithms based on variational [18] or Monte-Carlo sampling methods [19] may be used. In the case where all BN variables are observable and CPDs are in the exponential family, as in the HMM/BN model, the Maximum Likelihood (ML) parameter estimation algorithm may be used.

One problem specific to HMM/BN that may occur during BN training is the increased parameter number and consequently poor parameter estimation. Depending on the number and the size of the BN variables, encoding their conditional probabilistic dependencies may require quite a lot of parameters. Except for the rare cases when the amount of available training data is large enough, some clustering and parameter-tying schemes should be applied to reduce the actual number of parameters that need to be estimated.

During recognition, the only difference between traditional HMM and HMM/BN is that instead of calculating the state output as a Gaussian mixture, in the HMM/BN case, we infer it from the state BN. Depending on its topology and the types of its variables, the BN inference may be simple or quite complex. In this paper we show that under some conditions and by putting some limitations on the variables' types, the BN inference complexity can be reduced to that of a mixture of Gaussians. This makes the HMM/BN model computationally equivalent to the conventional HMM and, moreover, allows for a direct replacement of the HMM acoustic model with an HMM/BN model having the same state structure.

In this paper, we describe two example applications of the HMM/BN model for an LVCSR task [11], [20]. In the first case, as additional state BN variables we use the speaker gender and F0 frequency value. In the second case, the BN is designed to learn the correlation between neighboring speech frames.

The remainder of the paper is organized as follows. The next section gives a brief introduction of the HMM/BN model and provides details about its design, training and implementation. Section 3 provides details about the example systems based on the HMM/BN and conclusions are drawn in Sect. 4.

2. Hybrid HMM/BN Model

2.1 Background

The HMM/BN model is a combination of an HMM and a Bayesian Network. Speech temporal 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.

Structurally, the HMM/BN model is analogous to the hybrid HMM/NN model [21]. The difference is that instead of a Neural Network, the HMM is coupled with a BN. The HMM states in Fig. 1 share the same BN, which means that their probability density functions are the same. In some cases, it might be advantageous to have different BN topologies for different sets of states[†]. For example, first states of all HMMs may have different state BNs than center states or last states. Such cases are regarded as an extension of the basic HMM/BN model and all the methodology described below can be applied to them without a loss of generality.

By definition, a Bayesian Network represents a joint probability distribution of a set of random variables $Z_1, \ldots, Z_N$, 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 [22]:

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

In practice, the HMM state distribution is often modeled with a mixture of Gaussian functions. This can be graphically represented by a BN with topology shown in Fig. 2, where $M = \{m_j\}, j = 1, \ldots, K$ is a discrete variable representing mixture component index. Since the variable M is hidden, the data likelihood $p(x_t|q_i)$ can be calculated

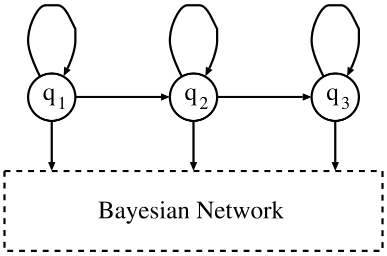

Fig. 1 HMM/BN model structure. HMM transitions model speech temporal characteristics and BN represents states' probability distributions.

[†] Here, we assume sets of states whose union consists of all the states of an acoustic model, not just a single HMM.

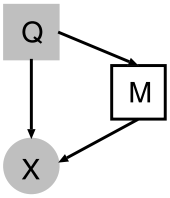

Fig. 2 BN representing mixture of Gaussians.

using the BN joint probability function (Eq. (1)) as follows:

$$\begin{aligned} p(x_t|q_i) &= \frac{P(x_t, q_i)}{P(q_i)} = \frac{\sum_{j=1}^{K} P(x_t, m_j, q_i)}{P(q_i)} \\ &= \frac{\sum_{j=1}^{K} P(x_t|m_j, q_i)P(m_j|q_i)P(q_i)}{P(q_i)} \\ &= \sum_{j=1}^{K} P(m_j|q_i)P(x_t|m_j, q_i). \end{aligned} \tag{2}$$

If we replace $P(m_j|q_i)$ with w_{ji} and $P(x_t|m_j, q_i)$ with Gaussian function $N(x_t; \mu_{ji}, \Sigma_{ji})$, we get a standard mixture of Gaussians equation:

$$p(x_t|q_i) = \sum_{j=1}^{K} w_{ji} N(x_t; \mu_{ji}, \Sigma_{ji}) \tag{3}$$

Figure 2 allows us to interpret the Gaussian mixture distribution in a different way. It shows that observation variable X depends not only on the state index but also on the variable M. However, M has no physical meaning. In this respect, Gaussian mixture learning is ”blind” and does not reflect the way a speech signal is produced; or at least it does not account for the factors it depends on, such as speaker gender, environmental noises, communication channels, etc. Variable M, for example, could represent pitch value, articulatory configuration or some other parameter that effects the speech spectrum.

2.2 HMM/BN Model Design and Training

The HMM/BN acoustic model design involves several main steps: choosing the speech unit to be modeled (phoneme, word, etc.); determining the number of states per unit and the state topology; and choosing the BN structure. The first two steps are essentially the same as for the standard HMM. Therefore, the same methods and techniques are applicable in the HMM/BN case. For small tasks and context-independent unit models, we need to set only an appropriate number of states: for monophones, usually three states and five or more for syllable units. The number of states for word-level models will depend on the average word duration and is mainly between 10 and 20 states. The state structure is almost always left-to-right. When a context-dependent (for example: triphone) unit model is required, traditional state clustering and tying techniques can be applied. That can be either phonetic tree clustering [23] or successive state splitting [24] algorithms.

Ideally, the BN structure should be learned automatically from the training data, but this is a very difficult task [13] and, usually, BN topology is chosen manually by taking into account the available data and the task at hand. The BN can have many variables corresponding to different speech features or variability factors. Dependencies are usually set according to prior knowledge or data correlation analysis. In this way, we can impose knowledge-based structure on the speech generation process and achieve a more precise speech model. Which BN variables should be hidden or observable depends on the available additional speech training data (pitch, articulatory observations, prosodic features, etc.) or high-level knowledge (speaker gender, environment factor, phoneme position, etc.). In case we do not have observations of some variable, we could assume it hidden. However, as in the Gaussian mixture example described in the previous section, in such cases, the training with the EM algorithm is “blind” and there is nothing to force the hidden variable to keep its “meaningful” interpretation, i.e., to represent a particular speech feature. Therefore, it is better to avoid having hidden BN variables during training.

As in the case of the HMM/NN model, parameter learning of the HMM/BN is based on the Viterbi training paradigm and can be summarized in the following algorithm.

- Step 1. Initialization.
- Step 2. Viterbi alignment.
- Step 3. Update BN parameters.
- Step 4. Update HMM transition probabilities.
- Step 5. Stop or go to Step 2.

The initialization step involves setting initial values of the model parameters (transition probabilities and BN parameters) given that state structure and BN topology are decided in advance. Although random initialization is possible, in practice, to facilitate the training we first train a bootstrap HMM model and use its state structure and transition probabilities to initialize the HMM/BN. For the BN, any initial parameter values can be used. Thus, the main part of HMM/BN training becomes the BN parameter estimation. Since the state variable Q is observable, before BN training we need to obtain its values for each sample of X. This is done by the Viterbi alignment step. For BN parameter estimation, several methods are available. When all BN variables are observable, i.e., in the full observability case, the maximum likelihood (ML) approach to parameter estimation can be easily applied. In this case, given the training data $O = \{o_t\}, t = 1, \ldots, T$ where each o_t contains observations of all BN variables, $o_t = \{z_{1_t}, \ldots, z_{N_t}\}$, the log-likelihood function is [5]:

$$\begin{aligned} L &= \log \prod_{t=1}^{T} Pr(o_t|G) \\ &= \log \prod_{t=1}^{T} P(Z_1 = z_{1_t}, \ldots, Z_N = z_{N_t}) \end{aligned}$$

$$= \sum_{i=1}^{N} \sum_{t=1}^{T} \log P(Z_i|Pa(Z_i))|_{z_{1_t},\ldots,z_{N_t}} \tag{4}$$

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 CPD given its local data $\{o_t(Z_i, Pa(Z_i))\}$. When the CPD is tabular (represented by CPT), then the log-likelihood becomes:

$$L = \sum_{ijk} N_{ijk} \log \theta_{ijk} \tag{5}$$

where by definition $\theta_{ijk} = P(Z_i = k|Pa(Z_i) = j)$ and N_{ijk} is the number of times the event $(Z_i = k, Pa(Z_i) = j)$ was seen in the training set. The ML estimate of θ_{ijk} is:

$$\hat{\theta}_{ijk} = \frac{N_{ijk}}{\sum_{k'} N_{ijk'}} \tag{6}$$

For continuous nodes that have discrete parents, the CPD can be represented by Gaussian functions:

$$p(z_i|Pa(Z_i) = j) = N(z_i; \mu_j, \Sigma_j) \tag{7}$$

for which the ML parameter estimates are well known.

In a partially observed case, i.e. when some of the (discrete) variables are hidden, the Expectation-Maximization (EM) algorithm can be applied. After BN is trained and its parameters fixed, the HMM transition probabilities are re-estimated with the standard forward-backward algorithm. All of these steps are repeated until the convergence criterion is met. This can be an increase in data likelihood or simply a fixed number of iterations.

When the BN is fully observable, the most time consuming steps of the HMM/BN training are the Viterbi alignment (Step 2) and transition probabilities update (Step 4) each of which have similar complexity to the HMM forward-backward training step. However, in practice, Step 4 may be skipped if the initial values of the transition probabilities are good enough, i.e., when initialized from a bootstrap HMM. In terms of the memory required to store the model parameters, HMM/BN and HMM do not differ significantly as long as they have a similar number of Gaussians.

2.3 Implementation and Decoding

The decoding in HMM-based ASR systems is usually done in a frame-synchronous manner using the Viterbi algorithm. As the difference between HMM/BN and HMM is in the way the state output probability is calculated, the same decoding strategy can be applied. Depending on the BN complexity and the type of its variables, the output probability inference can be done in different ways. In general, we need to obtain $P(x_t, z_t^1, \ldots, z_t^N|q_t)$, where x_t is the speech spectrum observation, $z_t^1, \ldots, z_t^N$ are the instances of all additional observable variables, and q_t is the HMM state ID at time t. This requires a BN inference engine that should be coupled with the Viterbi decoder and a feature extraction modules that will provide the $z_t^1, \ldots, z_t^N$ observations. Since this may not always be practical and in order to reduce the implementation costs, we can assume all additional variables hidden during recognition. This is especially useful when during recognition the extraction of some features is difficult or even impossible, like in the case of articulatory features.

A further simplification can be achieved if all additional BN variables are chosen to be discrete. In that case, for an arbitrary BN having joint pdf $P(X, Q, Z_1, \ldots, Z_N)$, from Eqs. (1) and (2), we have:

$$P(x|q)$$
$$= \sum_{z_1} \cdots \sum_{z_N} \prod_{i=1}^{N} P(z_i|Pa(z_i))P(x|Pa(x)). \tag{8}$$

Since all z_i and their parents are discrete, the product $\prod_{i=1}^{N} P(z_i|Pa(z_i))$ is a scalar and can be calculated easily. The probability $P(x|Pa(x))$ is usually represented by a Gaussian function and therefore the above equation represents a Gaussian mixture. The mixture weights, i.e. $\prod_{i=1}^{N} P(z_i|Pa(z_i))$, can be calculated in advance because they do not depend on x. Thus, each state is associated with a set of Gaussians and their weights. Note that in contrast to the conventional HMM[†], different states may have mixtures of different sizes depending solely on the BN variables' joint distribution given the state ID.

Such simplification of the BN inference is practically useful because the HMM/BN model becomes computationally equivalent to the HMM; therefore, there is no need for an inference engine or any modifications to the traditional HMM decoder.

Hiding the additional BN variables during recognition has another advantage for dealing with the "unseen data" problem that often occurs with limited training data. It is not unusual that some "contexts" (in terms of BN, that would be some combination of BN variables' values) may not appear in the training data, but be present in the test data. In such cases, if prior knowledge of "contexts" similarities is available, models for the "unseen data" can be generated. Otherwise, small fixed probabilities are assigned to all the problematic "contexts." In the BN case, when the output probability is calculated by Eq. 8, the problem is solved automatically because this equation, in fact, "marginalizes out" all the hidden variables.

2.4 Reduction of HMM/BN Parameter Number

The HMM/BN model is fully described by two sets of parameters: HMM transition probabilities and state BN parameters. The number of transition probabilities is proportional to the number of states and as it is usually in the order of thousands, their estimation is not a problem. For the BN, however, depending on the number of variables and their sizes, the parameter number may become too large,

[†] Although there are techniques that attempt to optimize the number of Gaussians, such as [25], the common approach is to use the same manually set mixture component number for each state.

making estimation quite poor. For example, let us consider the BN from Fig. 2 and assume that the additional variable M is not the Gaussian component ID, but rather some observable discrete speech variability factor. Then according to Eq. 2, there will be K (Gaussian) components $P(x|m_j, q_i)$, $j = 1, \ldots, K$ for each state q_i. If our HMM/BN acoustic model has several thousand states and the size of M is about one hundred, then the total number of Gaussians in the model will be in the order of hundreds of thousands.

Obviously, we need a huge database in order to train such a large model. In practice, however, the actual number of mixture components per state rarely reaches K. There are two reasons for that: 1) the conditional distribution $P(m|q)$ may have many zeroes. For example, if q_i represents an unvoiced speech interval and m_j denotes some pitch frequency, then, naturally, $P(m_j|q_i) = 0$; 2) there are no samples in the training data corresponding to some $P(m_j|q_i)$. This is the so-called "unseen data" phenomenon, and its handling was discussed in the previous section. Nevertheless, even though the number of mixture components per state does not reach K, we can still face the limited data problem since many Gaussians may have only a few training samples. The most widely used solutions are data clustering and parameter tying.

Since the BN is trained on several speech feature sets (spectral features as well as features represented by the additional BN variables), clustering can be applied to some or all feature sets depending on the required degree of parameter number reduction. By clustering the additional (discrete) features we effectively decrease the size of the state mixtures (K), which may reduce the feature space resolution and result in a less precise model. Clustering and tying of the Gaussians is more flexible because the mixtures can share some components but keep their original size, and more importantly, their original weights.

3. HMM/BN Application Examples

3.1 LVCSR System Using Pitch and Gender Information

To achieve high performance and manageable model size, most of the large-vocabulary speech recognition systems are based on context-dependent sub-word unit HMMs with a tied state structure. Our system is built in a similar manner, but instead uses crossword triphone HMM/BN models. The state-level topology (three-state left-to-right) and the tying scheme are taken from a bootstrap conventional HMM acoustic model trained on the same data.

Training LVCSR system requires a lot of data and the available databases of sufficient size consist of speech data only. Therefore, any speech feature other than a spectrum representation (MFCC, for example) that we would like to use as an additional BN variable should also be extracted from the speech signal. The fundamental frequency, or pitch as it is often called, is one such feature that can be obtained relatively easily from the speech waveform. Also, most databases contain information about the speaker's ID and gender. Therefore, it is easy to obtain a label for the speaker's gender that can be considered as a high-level discrete speech feature. Thus, we can use both pitch and gender as additional state BN variables. The structure of the state Bayesian Network we used is shown in Fig. 3, where variables Q and X represent the state and the speech spectrum feature and the other two - F and G - correspond to the pitch frequency and speaker gender. As can be seen, pitch depends on the speaker gender and they both influence the speech spectrum (X). Also, pitch as well as the speech spectrum depend on the phonetic unit state represented by Q. All the BN dependencies are set according to our prior knowledge about the relationship between the speech features [11].

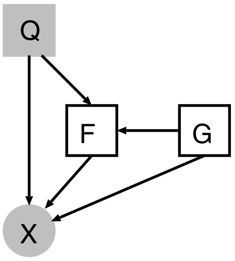

Fig. 3 State BN structure with pitch frequency F and speaker gender G as additional variables.

To make use of the simplified BN inference as described in Sect. 2.3, a discrete representation of the pitch frequency is necessary. For that, we used Vector Quantization. The discretization of a continuous variable always introduces some information loss, but in this case it offers the advantage of consistent representation of both voiced and unvoiced speech frames. This is possible if we consider the unvoiced frames as having zero pitch frequency and label them with identical labels.

During recognition, the speaker gender is not known, so the variable G is hidden. The pitch variable can be considered hidden as well despite the possibility of observing it, i.e. using a voiced/unvoiced detector, extracting pitch frequency and quantizing it. Under these conditions, the state output probability can be calculated from the BN joint pdf and Eq. (1) as:

$$\begin{aligned} P(x_t|q_t) &= \frac{P(x_t, q_t)}{P(q_t)} \\ &= \frac{\sum_{f,g} P(x_t, f, g, q_t)}{P(q_t)} \\ &= \frac{\sum_{f,g} P(x_t|f, g, q_t)P(f|g, q_t)P(g)P(q_t)}{P(q_t)} \\ &= \frac{1}{2}\sum_{f,g} P(f|g, q_t)P(x_t|f, g, q_t). \end{aligned} \tag{9}$$

In the above equation, we assume that the prior probabilities of the speaker being male or female are equal, so $P(g) = 0.5$.

We evaluated our HMM/BN-based LVCSR system using the WSJ database. The experimental setup followed closely the HUB2 (Nov93) evaluation specifications [26]. For training we employed the SI-284 training set. The language model was a standard bigram provided for the HUB2 evaluation. The test set consisted of 215 utterances with 0% OOV and a 5,000-word dictionary.

Speech data were transformed into 39 dimensional feature vectors (Pow + 12MFCC + ΔPow + 12ΔMFCC + $\Delta\Delta$Pow + 12$\Delta\Delta$MFCC) from 20-ms long frames with 10-ms shift. The pitch frequency was obtained from the speech signal by Entropic's ESPS package which uses the pitch tracking algorithm described in [27]. The pitch extraction rate was the same as for the speech features, so for each cepstrum vector there was a corresponding pitch value. Zero pitch was set for silence and non-voiced frames. From all non-zero pitch data two VQ codebooks were trained with three and seven centroids respectively. Later, a zero centroid was added manually to each of the codebooks, so the number of centroids became four and eight. The pitch data were then quantized and codebook labels were obtained. Thus, each speech feature vector was labeled with a pitch and speaker gender label.

Using the HTK speech toolkit we trained three tied-state crossword triphone bootstrap HMM models with 10,071, 7,870 and 5,666 states respectively. They were used to initialize three HMM/BN models. State labels for the first iteration of the HMM/BN training were obtained from the bootstrap models by Viterbi alignment. Because all the BN variables were observable (for the training), BN parameters were estimated using the ML algorithm. During training, each Gaussian $p(x|f, g, q)$ was estimated from its local data, i.e., from speech vectors labeled with the same gender, pitch and state labels. In some cases, the amount of this data was too sparse and in order to avoid badly trained parameters, we used thresholding and Gaussian tying. A threshold $trh = 100$ was applied to each Gaussian's local data (vector) number and if it was below the threshold, the data were pooled together with those that have the same state and pitch labels but the opposite gender label. If the pooled data vector number exceeded the threshold, the two Gaussians were tied, otherwise they were removed from the model. The number of HMM/BN model training iterations was set to five.

Table 1 and Table 2 respectively show the results using four and eight level quantized pitch data. In the HMM/BN case, since the mixture number varies from state to state, the average number of Gaussians per state is given. For comparison, results for a similarly complex HMM model are shown.

For the case of four-level quantized pitch data and a lesser number of model parameters, we obtained better results than the baseline HMM. Furthermore, the relative improvement was highest for the model with smallest state number. On the other hand, with eight level quantized data, HMM/BN did not improve the baseline HMM performance, but for the case of the smallest state number, WERs were almost the same. This indicates that the amount of training data may not be sufficient in the eight-level CB case. Also, since the pitch extraction is not error-free, quantization with a CB of a larger size makes the BN parameter estimation more sensitive to such errors.

Table 1 Results using 4 level CB quantized pitch data.

Model	states	mix/state	WER (%)
HMM	10071	4	12.4
HMM/BN		3.7	11.8
HMM	7850	4	14.7
HMM/BN		4.1	14.0
HMM	5666	5	13.6
HMM/BN		4.5	12.4

Table 2 Results using 8 level CB quantized pitch data.

Model	states	mix/state	WER (%)
HMM	10071	6	11.2
HMM/BN		5.6	12.1
HMM	7850	6	13.3
HMM/BN		5.9	13.8
HMM	5666	7	12.5
HMM/BN		6.6	12.8

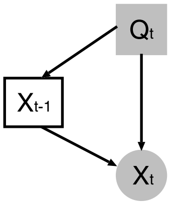

Fig. 4 BN topology for modeling dependency on the previous observation.

3.2 Modeling Successive Frame Dependencies

The advantage of using the BN as a state distribution model is that it is very easy to add additional variables. In order to model the dependency between the current and previous observations, we can add one additional variable representing speech spectrum feature at time $t-1$ as shown in Fig. 4.

Variables x_t and x_{t-1} take real values and there are several choices for modeling their conditional distribution, such as, for example, a Linear Regression (LR) model [28]. However, when the number of states is too big, as in the context-dependent acoustic models, having an LR or any other complex CPD approximation increases the complexity of the entire model. The approach we took is to convert x_{t-1} into a discrete variable by means of VQ. This simplifies the HMM/BN implementation as described in Sect. 2.3 and is similar to the previous example where we used discrete pitch values. Since all BN variables are observable during training, the BN is trained by ML algorithm. During recognition, the x_{t-1} variable can be either observable (labels can be obtained by VQ) or hidden. The latter case, however, simplifies the $P(x_t|q_t)$ inference to a Gaussian mixture calculation. Similarly to Eq. (2), we have:

$$P(x_t|q_t) = \sum_{x_{t-1}} P(x_{t-1}|q_t)P(x_t|x_{t-1}, q_t) \qquad (10)$$

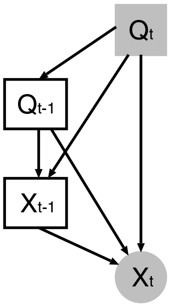

Fig. 5 BN topology for modeling dependency on both previous observation and previous state.

where summation is done over all discrete values of x_{t-1}.

Extending this model to include the current observation's dependency on the previous state is as easy as adding another discrete variable representing q_{t-1}. The BN topology in this case is shown in Fig. 5. Assuming again that x_{t-1} and q_{t-1} are hidden during recognition, for $P(x_t|q_t)$ we get:

$$\begin{aligned} &P(x_t|q_t) \\ &= \sum_{q_{t-1}} \sum_{x_{t-1}} P(q_{t-1}|q_t) P(x_{t-1}|q_{t-1}, q_t) \\ &\quad P(x_t|x_{t-1}, q_{t-1}, q_t) \end{aligned} \tag{11}$$

where the double sum is over all state IDs and all values of x_{t-1}.

For acoustic model training, we used the same WSJ-284 data as in the previous example. However, in this case, we used different test data that consisted of 200 utterances selected randomly from a set of 4000 read speech utterances spoken by 40 speakers. The speech material of the test data consists of travel-related expressions and is quite different from that of the training data. All speech utterances were collected in quiet environments. Here, 25-dimensional (12MFCC + 12ΔMFCC + Pow) feature vectors are extracted with a 20 ms sliding window at a 10 ms frame rate. The language model used in these experiments was a word bi-gram and was trained on a different text corpus consisting of about 150,000 travel domain sentences. Note that the WSJ sentences come from financial news articles. The vocabulary size was about 20,000 words and the test data out-of-vocabulary rate was about 1.5%.

Our baseline acoustic model is an HMnet obtained by a successive state splitting algorithm with an MDL stopping criterion [29]. The total number of states is 2009. Four versions with 5, 10, 15 and 20 Gaussian components per state were trained in order to compare models with different parameter numbers. The HMM/BN models were initialized using the baseline HMnet, meaning that they have the same number of states and the same state topology.

As can be seen from Eqs. (10) and (11), the number of Gaussian components of the HMM/BN model depends on the VQ codebook size, and this number can become quite large. Indeed, using the BN topology of Fig. 4 with a VQ size of 128 resulted in more than 100,000 Gaussians. To reduce the parameter number, we applied Gaussian clustering and tying as described in Sect. 2.4. In this way, for each VQ codebook size of 32, 64 and 128, we made four models, with the total number of Gaussians corresponding to that of the baseline models, i.e., models with 5, 10, 15 and 20 mixture components per state in average.

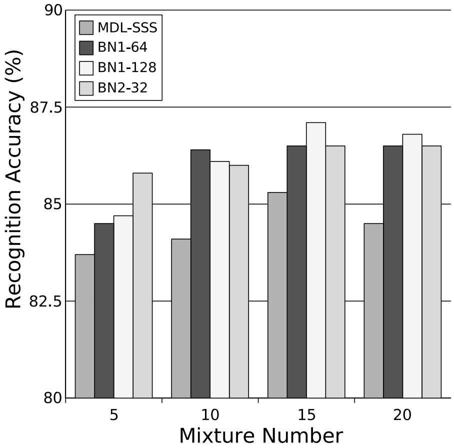

Fig. 6 Results of LVCSR experiments.

We evaluated HMM/BN models having all possible variants of BN structures and codebook sizes and the results of the best three and the baseline MDL-SSS HMnet are shown in Fig. 6. In this figure, the HMM/BN with the BN topologies from Fig. 4 and Fig. 5 are denoted as BN1 and BN2, respectively. The numbers after the name indicate the size of the VQ codebook used. The improved performance of the HMM/BN model clearly shows that it was able to utilize the frame correlation information effectively.

4. Conclusion

In this paper, we described the hybrid HMM/BN model and discussed some issues related to its design and implementation. Although this model can be regarded as a pure Bayesian Network, its structure allows simple algorithms to be used for training and recognition instead of general BN learning and inference methods which depending on the task may often become computationally intractable.

Since the HMM/BN has the same state topology as the HMM, the way we build acoustic models is not altered at all. The only difference is the need for BN training, which in many cases can be reduced to an easy ML parameter estimation. The implementation of the HMM/BN can be simplified by forcing all the additional BN variables to be discrete and assuming them hidden during recognition. This way, the BN inference becomes equivalent to a Gaussian mixture computation.

As the provided examples of HMM/BN application show, even with a few additional variables and simple BN topologies, the hybrid model achieved better performance than the conventional HMM.

Acknowledgment

The research reported here was supported in part by the Ministry of Public Management, Home Affairs, Posts and

Telecommunications of Japan.

References

[1] S.E. Levinson, L.R. Rabiner, and M.M. Sondhi, "An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition," Bell Syst. Tech. J., vol.62, no.4, pp.1035–1074, April 1983.

[2] L.R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," Proc. IEEE, vol.77, no.2, pp.257–285, Feb. 1989.

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

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

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

[6] K. Daoudi, D. Fohr, and C. Antoine, "A new approach for multiband speech recognition based on probabilistic graphical models," Proc. ICSLP, pp.329–332, 2000.

[7] T. Shinozaki and S. Furui, "Dynamic Bayesian network-based acoustic models incorporating speaking rate effects," IEICE Trans. Inf. & Syst., vol.E87-D, no.10, pp.2339–2347, Oct. 2004.

[8] T. Stephenson, M. Mathew, and H. Bourlard, "Modeling auxiliary information in Bayesian network based ASR," Proc. Eurospeech, pp.2765–2768, 2001.

[9] A. Garg, V. Pavlovic, and J. Rehg, "Audio-visual speaker detection using dynamic Bayesian networks," Proc. Int. Conf. on Automatic Face and Gesture Recognition, pp.384–390, 2000.

[10] K. Markov and S. Nakamura, "Modeling HMM state distributions with Bayesian networks," Proc. ICSLP, pp.1013–1016, 2002.

[11] K. Markov and S. Nakamura, "Hybrid HMM/BN LVCSR system integrating multiple acoustic features," Proc. ICASSP, pp.888–891, 2003.

[12] K. Markov, J. Dang, Y. Iizuka, and S. Nakamura, "Hybrid HMM/BN ASR system integrating spectrum and articulatory features," Proc. Eurospeech, pp.965–968, 2003.

[13] D. Heckerman, "A tutorial on learning with Bayesian networks," in Learning in Graphical Models, ed. M. Jordan, pp.301–354, MIT Press, 1999.

[14] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, San Mateo, California, 1988.

[15] M. Peot and R. Shachter, "Fusion and propagation with multiple observations," Artif. Intell., vol.48, no.3, pp.299–318, 1991.

[16] F. Jensen, S. Lauritzen, and K. Olsen, "Bayesian updating in recursive graphical models by local commutations," Comput. Stat. Data Anal., vol.4, pp.269–282, 1990.

[17] F. Jensen, An Introduction to Bayesian Networks, UCL Press, London, 1996.

[18] M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul, "An introduction to variational methods for graphical models," in Learning in Graphical Models, ed. M. Jordan, pp.105–161, MIT Press, 1999.

[19] D. McKay, "Introduction to Monte Carlo methods," in Learning in Graphical Models, ed. M. Jordan, pp.175–204, MIT Press, 1999.

[20] K. Markov and S. Nakamura, "Modeling successive frame dependencies with hybrid HMM/BN acoustic model," Proc. ICASSP, pp.701–704, 2005.

[21] H. Bourlard and N. Morgan, Connectionist Speech Recognition: A Hybrid Approach, Kluwer Academic Publishers, Boston, 1994.

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

[23] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valchev, and P. Woodland, The HTK Book, Cambridge University Engineering Department, 2002.

[24] J. Takami and S. Sagayama, "A successive state splitting algorithm for efficient allophone modeling," Proc. ICASSP, pp.573–576, 1992.

[25] S. Chen and P. Gopalakrishnan, "Clustering via the Bayesian information criterion with applications in speech recognition," Proc. ICASSP, pp.645–648, May 1998.

[26] Advanced Research Projects Agency, Proceedings of the Spoken Language Technology Workshop, Plainsboro, New Jersey, Morgan Kaufmann Publishers, March 1994.

[27] D. Talkin, "A robust algorithm for pitch tracking (PART)," in Speech Coding and Synthesis, ed. W. Kleijn and K. Paliwal, Elsevier, New York, 1995.

[28] J. Bilmes, "Buried Markov models: A graphical-modeling approach to automatic speech recognition," Comput. Speech Lang., vol.17, no.2-3, pp.213–231, 2003.

[29] 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, pp.2121–2129, Aug. 2004.

Konstantin Markov was born in Sofia, Bulgaria. After graduating with honors from the St. Petersburg Technical University, he worked for several years as a research engineer at the Communication Industry Research Institute in Sofia. He received his M.Sc. and Ph.D. degrees in electrical engineering from Toyohashi University of Technology, Japan, in 1996 and 1999, respectively. In 1998, he received the Best Student Paper Award from the IEICE Society. In 1999, he joined the research development department of ATR, Japan, and in 2000 became an invited researcher at the ATR Spoken Language Communication (SLC) Research Laboratories. Currently, he is a senior research scientist at the Acoustics and Speech Processing Department of ATR SLC. He is a member of ASJ, IEEE and ISCA. His research interests include signal processing, automatic speech recognition, Bayesian networks and statistical pattern recognition.

Satoshi Nakamura was born in Japan on August 4, 1958. He received his B.S. degree in electronic engineering from Kyoto Institute of Technology in 1981 and his Ph.D. degree in information Science from Kyoto University in 1992. Between 1981–1993, he worked with the Central Research Laboratory, Sharp Corporation, Nara, Japan. During 1986–1989, he worked with ATR Interpreting Telephony Research Laboratories. From 1994–2000, he was an associate professor of the graduate school of Information Science, Nara Institute of Science and Technology, Japan. In 1996, he was a visiting research professor of the CAIP center of Rutgers University of New Jersey USA. He is currently the head of the Acoustics and Speech Research Department at ATR Spoken Language Translation Laboratories, Japan. He also has served as an honorary professor of University Karlsruhe, Germany from 2004. His current research interests include speech recognition, speech translation, spoken dialog systems, stochastic modeling of speech, and microphone arrays. He received the Awaya Award from the Acoustical Society of Japan in 1992, and the Interaction2001 Best Paper Award from the Information Processing Society of Japan in 2001. He served as an associate editor for the Journal of the IEICE Information from 2000–2002 and is currently a member of the Speech Technical Committee of the IEEE Signal Processing Society. He is a member of the Acoustical Society of Japan, Information Processing Society of Japan, and IEEE.