Continuous Speech Recognition Using Dynamic Bayesian Networks : A Fast Decoding Algorithm

Murat Deviren, Khalid Daoudi
INRIA-LORIA, Speech Group
B.P. 101 - 54602 Villers les Nancy, France
deviren,daoudi@loria.fr

Abstract

State-of-the-art automatic speech recognition systems are based on probabilistic modelling of the speech signal using Hidden Markov Models (HMMs). Recent work has focused on the use of dynamic Bayesian networks (DBNs) framework to construct new acoustic models to overcome the limitations of HMM based systems. In this line of research we proposed a methodology to learn the conditional independence assertions of acoustic models based on structural learning of DBNs. In previous work, we evaluated this approach for simple isolated and connected digit recognition tasks. In this paper we evaluate our approach for a more complex task: continuous phoneme recognition. For this purpose, we propose a new decoding algorithm based on dynamic programming. The proposed algorithm decreases the computational complexity of decoding and hence enables the application of the approach to complex speech recognition tasks.

1 Introduction

One of the basic building blocks of a speech recognition system is the acoustic modelling of sub-speech units. (i.e. words, phones, biphones, etc.). The accuracy of acoustic modeling is a key point in the performance of a recognition system. The most popular probabilistic models used for this task are Hidden Markov models (HMMs). Recently, there has been an increasing interest in a more general class of probabilistic models which include HMM only as a special case: *dynamic Bayesian networks* (DBNs). An important and attractive property of the DBNs formalism is the ability to encode complex dependency relations. In the last few years there has been several approaches to exploit this property in speech recognition systems (Bilmes, 2000), (Daoudi et al., 2001b), (Daoudi et al., 2001a), (Cetin et al., 2002), (Zweig, 1998), (Zweig et al., 2002). These provide several useful capabilities that are not readily achievable by HMM because of its restrictive Conditional Independence (CI) assertions.

In (Deviren and Daoudi, 2001) we proposed a methodology to learn the CI assertions of acoustic models based on structural learning of DBNs. As in HMMs, we consider that the observations are governed by a hidden process. On the other hand, we do not make any *a priori* dependency assumption between the hidden and observed processes. Rather, we give data a relative freedom to dictate the appropriate dependencies. In other words, we learn the dependencies between (hidden and observable) variables *from data*. The principle of this methodology is to search over all possible "realistic" dependencies, and to choose the ones which best explain the data using the MDL score. This approach has the advantage to *guarantee* that the resulting model represents speech with higher fidelity than HMMs. Moreover, a *control* is given to the user to specify the maximal dependency structure and hence to make a trade-off between modeling accuracy and model complexity. The proposed approach is also technically very attractive because all the computational effort is performed in training phase.

We presented the application of this approach

to isolated speech recognition in (Deviren and Daoudi, 2001). In (Deviren and Daoudi, 2002) we extended our approach for continuous speech recognition and presented preliminary results on a connected digit recognition task. However the applicability of the decoding algorithm used in (Deviren and Daoudi, 2002) is limited because of its computational complexity. In this paper, we provide in particular a much faster decoding algorithm which handles different model structures for different acoustic units. The algorithm is based on a dynamical programming procedure and leads to a substantial gain in complexity as compared to the algorithm in (Deviren and Daoudi, 2002).

In our previous contributions, we evaluated our methodology on simple databases and tasks. In this paper, we evaluate our system on a phone recognition task using the TIMIT database (which is ideal for phone recognition (Lee and Hon, 1989)). We provide a comparison of our system with an HMM based system using the HTK toolkit [1]. The results show that our methodology of learning the model dependencies from data still leads to significant improvement w.r.t. standard HMM modeling.

In the next section, we introduce the DBNs terminology. In section 3, we define the class of DBNs we use in our setting. We then briefly summarize the structural learning algorithm and our training strategy. In section 5, we describe a fast decoding algorithm for continuous speech recognition using the set of DBNs we consider. Finally, we illustrate the performance of our approach on a phone recognition task using the TIMIT database.

2 Dynamic Bayesian Networks

Our approach is based on the framework of *dynamic Bayesian networks* (DBNs). DBN theory is a generalization of Bayesian network (BN) theory to dynamic processes. Briefly, the BNs formalism consists of associating a directed acyclic graph to the joint probability distribution (JPD) $P(X)$ of a set of random variables $X = \{X_1, ..., X_n\}$. The nodes of this graph represent the random variables, while the arrows encode the conditional independencies (CI) which (are supposed to) exist in the JPD. The set of all CI relations, which are implied by the separation properties of the graph, are termed the *Markov properties*. A BN is completely defined by a graph structure S and the numerical parameterization Θ of the conditional probabilities of the variables given their parents. Indeed, the JPD can be expressed in a factorized form as $P(X) = \prod_{i=1}^{n} P(X_i|\Pi_i)$, where Π_i denotes the parents of X_i.

A DBN encodes the joint probability distribution of a time evolving set $X[t] = \{X_1[t], \ldots, X_n[t]\}$ of variables. If we consider T time slices of variables, the DBN can be considered as a (static) BN with $T \times n$ variables. Using the factorization property of BNs, the JPD of $\mathbf{X}_1^T = \{X[1], \ldots, X[T]\}$ can be written as :

$$P(X[1], \ldots, X[T]) = \prod_{t=1}^{T} \prod_{i=1}^{n} P(X_i[t]|\Pi_{it}) \quad (1)$$

where Π_{it} denotes the parents of $X_i[t]$. In the BNs literature, DBNs are defined using the assumption that $X[t]$ is a Markov process (Friedman et al., 1998). In this paper, we do not make such an assumption in order to allow non-Markov processes. By doing so our models are able to take into account, for instance, the well known *anticipation* phenomenon which occurs in the speech production mechanism. Precisely, we consider that the process $X[t]$ satisfies:

$$P(X_i[t]|\mathbf{X}_1^{t+\tau_f}) = P(X_i[t]|X[t-\tau_p], \ldots, X[t+\tau_f]) \quad (2)$$

for some positive integers τ_p and τ_f. Graphically, the above assumption states that a variable at time t can only have parents in the interval $[t-\tau_p, t+\tau_f]$. However, care must be taken when dealing with boundary variables. Namely, for each of the first τ_p and the last τ_f time slices the dependency time window is

[1] The Hidden Markov Model Toolkit (HTK) is a portable toolkit for building and manipulating hidden Markov models. It is widely used in the speech community and generally considered as a reference system for HMM based speech recognition. See http://htk.eng.cam.ac.uk/ for details on the toolkit.

different. In these time slices, the local structure of the DBN is different than the repeating transition structure. Taking this into account, the DBN is represented with $(\tau_p + \tau_f + 1)$ static BNs: τ_p initial networks, τ_f final networks and a transition network. (see (Deviren and Daoudi, 2001) for details). The associated conditional probabilities for each of these networks should be defined separately.

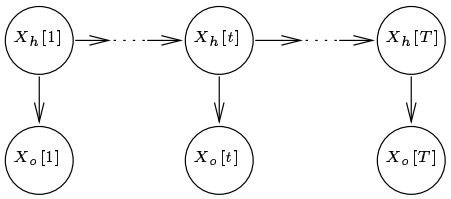

Figure 1: *HMM represented as a DBN.* $X_h[t]$ *denotes the hidden state variable at time* t *and* $X_o[t]$ *denotes the observation at time* t.

From this perspective, it is easy to represent an HMM as a DBN. Indeed, the Markov properties (dependency semantics) of an HMM, are encoded by the graphical structure shown in Fig. 1. Each node in this structure represents a random variable $X_h[t]$ or $X_o[t]$, whose value specifies the state or the observation at time t.

3 Structure Search Class

Our goal is to learn from data the DBN structures (and their corresponding parameters) which "best" explain data, i.e., the "appropriate" dependencies between the hidden and observed variables (the $X_h[t]$ and $X_o[t]$). This requires a search over a class of structures. Searching over all the possible DBN structures would be computationally infeasible. Therefore, we restrict ourselves to a small but rich set of structures that represents only *realistic* dependencies, in a physical and computational sense. The reader is referred to (Deviren and Daoudi, 2001) for the reasoning behind the authorized dependencies. These are formally defined as follows.

Let $X[t] = \{X_h[t], X_o[t]\}$ be the set of hidden and observed variables at time t. The allowed dependencies are :

- The hidden variable at time t is independent of $\mathbf{X}_1^{t-\kappa-1}$ given the last κ hidden variables, for $t > \kappa$,

$$P(X_h[t]|\mathbf{X}_1^{t-1}) = p(X_h[t]|X_h[t-\kappa], \ldots, X_h[t-1]). \quad (3)$$

- The observation variable at time t is independent of all other variables given the hidden variables in the time window $[t - \tau_p, t + \tau_f]$, for some positive integers τ_p and τ_f,

$$P(X_o[t]|\mathbf{X}_1^T \setminus \{X_o[t]\}) = p(X_o[t]|X_h[t-\tau_p], \ldots, X_h[t+\tau_f]). \quad (4)$$

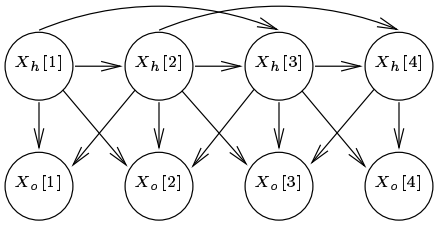

Figure 2: *DBN structure with* $(\kappa, \tau_p, \tau_f) = (2, 1, 1)$, $T = 4$

Hence, the search class of allowed DBN structures is defined by the triples (κ, τ_p, τ_f) for, $1 \leq \kappa \leq \kappa_{max}$, $0 \leq \tau_p \leq \tau_{p_{max}}$, $0 \leq \tau_f \leq \tau_{f_{max}}$, where $(\kappa_{max}, \tau_{p_{max}}, \tau_{f_{max}})$ is an upper bound which restricts the size of the search class. At the lower bound $(\kappa, \tau_p, \tau_f) = (1, 0, 0)$, the structure reduces to the standard first-order HMM (Figure 1) where, Eq.(3) defines the state transition probabilities and Eq.(4) defines the observation probabilities. Each triple (κ, τ_p, τ_f) within this interval determines completely a DBN structure. In this paper we use $(\kappa_{max}, \tau_{p_{max}}, \tau_{f_{max}}) = (2, 1, 1)$ as the upper bound on the structure search class. This structure is shown in Figure 2.

If each discrete hidden variable $X_h[t]$ takes its values in the set of ordered labels $I = \{1_v \ldots N_v\}$ [2], and each observable variable has a

[2]The subscript v is used to refer later to an acoustic unit v.

conditional Gaussian density, the numerical parameterization of the set of DBNs in our search class is the following:

$$\begin{aligned} P(X_h[t] = j | \Pi_{ht} = \mathbf{i}) &= a_{\mathbf{i}j}[t], \ for\ j \in I \\ P(X_o[t] | \Pi_{ot} = \mathbf{i}) &\sim \sum_k c_k \mathcal{N}(\mu_{\mathbf{i}_k}[t], \Sigma_{\mathbf{i}_k}[t]). \end{aligned}$$

The (possibly) vector-index $\mathbf{i}$ is over all possible values of the variable's parents. For the specific structure space defined, $\mathbf{i}$ is a point in the cartesian space I^m, where m is the number of parents of a variable.

4 Learning Algorithm

In this section, we describe our training strategy for continuous speech recognition based on context independent DBN phone models. In the previous section, we defined a set of plausible DBN structures for speech recognition. We use this set as the search class for our structural learning algorithm to find the "optimal" DBN structure and the associated parameters for each phone v in the given phone set V. The principle of the learning algorithm is to initialize each DBN with some initial structure and parameters and to run a generalized version of the Expectation Maximization (EM) algorithm, called structural EM (SEM (Friedman et al., 1998)) which allows the update of structure and parameters iteratively. The details of the algorithm is given in (Deviren and Daoudi, 2001) and the references within.

We initialize our DBNs with the HMM structure $(\kappa, \tau_p, \tau_f) = (1, 0, 0)$ which is the lower bound of our search space. This initialization guarantees that the resulting DBN will model speech with a higher (or equal) fidelity, as compared to HMMs. The trade off between the complexity of the learning algorithm and the fidelity of the resulting model is controlled by the upper bound on the search space.

The learning algorithm in (Deviren and Daoudi, 2002) requires a set of isolated observations for each DBN. However, in continuous speech recognition the training database consists of continuous utterances. Each utterance is labelled with the underlying phone sequence but in general the boundaries dividing the phone segments are either unavailable or unreliable. Therefore an embedded training strategy is necessary for reliable acoustic modeling. In that, we make use of a reference HMM system. First we train an HMM system using embedded training strategy. Next, we obtain the best segmentation of each training sentence using a forced alignment procedure with this reference system. We consider these segments as isolated observations for each phone and run the structural learning algorithm to obtain our DBN models. By doing so we rely on the segmentation provided by HMMs and improve the modeling accuracy within each segment.

5 Decoding Algorithm

Let us assume that we are given a set V of $|V|$ acoustic units, and a DBN model for each acoustic unit $v \in V$, with structure $(\kappa^v, \tau_p^v, \tau_f^v)$. The decoding problem is to identify the most likely sequence of these units, given a speaker utterance. In (Deviren and Daoudi, 2002) we present a decoding algorithm based on the use of Dawid's algorithm (Dawid, 1992) on a state-augmented DBN. Dawid's algorithm is a message propagation algorithm that allows the identification of the most likely sequence of hidden states in an arbitrary (discrete) Bayesian network given the observations (Dawid, 1992). Using our augmentation methodology we were able to represent all acoustic units' models in a single DBN. Hence, using Dawid's algorithm on this augmented DBN, one obtains the most likely sequence of acoustic units given the observations. The algorithm presented in (Deviren and Daoudi, 2002), however, is rather a "brute force" approach which introduces substantial unnecessary computations. In the following we present an equivalent[3] but much faster algorithm which takes advantage of the particular DBNs we are dealing with.

As in (Deviren and Daoudi, 2002), the first step is to construct a "maximal" DBN for each acoustic unit. The goal is to represent all acous-

[3] in the sense that it is an exact inference algorithm and yields the same decoding results as in (Deviren and Daoudi, 2002)

tic models with the same graphical structure without violating the learned Conditional Independence (CI) relations. The maximal network structure $(\kappa^m, \tau_p^m, \tau_f^m)$ is chosen such that all the learned CI relations can be encoded using this maximal structure. Precisely,

$$\kappa^m = \max_v(\kappa^v),\ \tau_p^m = \max_v(\tau_p^v),\ \tau_f^m = \max_v(\tau_f^v).$$

Then, the numerical parameterization of each maximal network is obtained by manipulating the one of the corresponding DBN, without violating its (learned) Markov properties. This is achieved by setting several conditional probabilities of the maximal DBN to be equal to those given by the learned DBN. Precisely,

$$P_m(X_i[t]|\Pi_{it}^m = (\mathbf{i},\mathbf{j})) = P(X_i[t]|\Pi_{it} = \mathbf{i}) \quad \text{for all } \mathbf{i} \in \Pi_{it}, \mathbf{j} \in \Pi_{it}^m \setminus \Pi_{it} \qquad (5)$$

where Π_{it}^m is the set of parents of $X_i[t]$ in the maximal DBN, and Π_{it} is the set of parents of $X_i[t]$ in the learned DBN.

Once the maximal network of each acoustic unit is constructed, the second step (which is the heart of this paper) consists, basically speaking, in operating a dynamical programming procedure to perform a "parallel" inference algorithm on "local clique potentials" (see (Kjaerulf, 1992) for details).

The principle of the algorithm is similar to Viterbi decoding algorithm that is widely used in HMM based speech recognition systems (Viterbi, 1967). The main difference is that Viterbi algorithm is used for first order hidden Markov models whereas the proposed algorithm is derived for the set of DBN structures we propose in (Deviren and Daoudi, 2001). Indeed, if the maximal structure is $(1,0,0)$, i.e. HMM structure, the algorithm is identical to the Viterbi decoding algorithm. We proceed now to describe this algorithm.

In the experiments we carry out later, the maximal network structure we obtain is $(\kappa^m, \tau_p^m, \tau_f^m) = (2,1,1)$. Given the technicality of the algorithm and for the sake of clarity and simplicity of the paper, we describe our decoding algorithm for this particular case. We emphasize however that this algorithm can be readily generalized to any kind of maximal network.

Let us denote the hidden and observed variables $X_h[t], X_o[t]$ as H_t and O_t respectively. However H_t takes now its values in $\{1_v, \ldots, N_v,\ \forall v \in V\}$ to denote the hidden states for all phone models. As described in Section 2, the conditional probabilities of a DBN are defined according to the initial, transition and final networks. For the maximal structure (2,1,1) there are 1 initial, 1 transition and 1 final networks. The conditional probabilities of these networks are denoted as follows (where $i,j,k \in \{1_v, \ldots, N_v,\ \forall v \in V\}$).

Initial network :

$$\begin{aligned} a_j^v &= P(H_1 = j) \\ a_{jk}^v &= P(H_2 = k|H_1 = j) \\ b_{jk}^{Iv}(O_1) &= P(O_1|H_1 = j, H_2 = k) \end{aligned}$$

Transition network :

$$\begin{aligned} a_{ijk}^v &= P(H_{t+1} = k|H_t = j, H_{t-1} = i) \\ b_{ijk}^v(O_t) &= P(O_t|H_{t-1} = i, H_t = j, H_{t+1} = k) \end{aligned}$$

Final network :

$$b_{ij}^{Fv}(O_t) = P(O_t|H_{t-1} = i, H_t = j)$$

Our aim is to find the most likely sequence of hidden states $H_1^T = \{H_1, \ldots, H_T\}$ given the observation sequence $O_1^T = \{O_1, \ldots, O_T\}$, and consequently the likelihood of observations along this sequence:

$$P_{max}(O_1^T) = \max_{H_1 \ldots H_T} P(O_1^T, H_1^T) \qquad (6)$$

First note that the state transitions are restricted to a left-to-right topology and a model transition from model r to v is only allowed from the last state of r (N_r) to the first state of v (1_v). In order to provide a recursive formulation for Eq. 6 we define two intermediate quantities:

$$\delta_t^v(i,j,k) = \max_{H_1 \ldots H_{t-2}} P(O_1^t, H_1^{t-2}, H_{t-1} = i, H_t = j, H_{t+1} = k)$$

$$\gamma_t^v(i,j) = \max_{H_1 \ldots H_{t-2}} P(O_1^t, H_1^{t-2}, H_{t-1} = i, H_t = j)$$

$\delta_t^v(i,j,k)$ is the likelihood of the first t observations along the most likely state sequence up to H_{t-2} and for $H_{t-1} = i, H_t = j, H_{t+1} = k$. Similarly $\gamma_t^v(i,j)$ is the likelihood of the first t observations along the most likely state sequence up to H_{t-2} and for $H_{t-1} = i, H_t = j$ assuming that O_t is the last observation from phone v. We need this second quantity to compute the likelihood when a phone transition occurs at $t+1$.

For each model v we initialize the recursion with $\delta_1^v(i,j,k) = a_j^v a_{jk}^v b_{jk}^{Iv}(O_1)$ which is the the likelihood of O_1 emitted from the initial network of v. Then for each t, the new evidence O_t can be emitted from either of the initial, transition and final networks of v. We know that $\delta_{t-1}^v(n,i,j)$ is the maximum likelihood for the observation sequence O_1^{t-1} for $H_{t-2} = n, H_{t-1} = i, H_t = j$.

If the emission is from the final network then O_t does not depend on H_{t+1} and the new likelihood is computed by maximizing $\delta_{t-1}^v(n,i,j)$ over n and incorporating the emission probability from the final network. We defined this term as $\gamma_t^v(i,j)$:

$$\gamma_t^v(i,j) = b_{ij}^{Fv}(O_t) \max_n [\delta_{t-1}^v(n,i,j)]$$

This term is the maximum likelihood of the observation sequence O_1^t for $H_{t-1} = i, H_t = j$ for $i,j \in \{1_v, \ldots, N_v\}$. Considering the fact that for the full observation sequence O_1^T the last observation is emitted from the final network of some $v \in V$, the maximum likelihood for the full observation sequence is obtained from the following maximization over v,i,j:

$$P_{max}(O_1^T) = \max_{i,j,v} [\gamma_T^v(i,j)]$$

To complete our formulation we continue with the derivation of the recursion formula for $\delta_t^v(i,j,k)$. If the emission is from initial or transition networks of v the incorporation of the new evidence O_t depends on the value of H_{t-2} such that:

- if $H_{t-2} \in \{1_v, \ldots, N_v\}$ then O_{t-1} was emitted from v and O_t will be emitted from the transition network of v. The new likelihood is computed by maximizing $\delta_{t-1}^v(n,i,j)$ over n and incorporating the emission probability from the transition network of v.

$$\delta_t^v(i,j,k) = a_{ijk}^v b_{ijk}^v(O_t) \max_n [\delta_{t-1}^v(n,i,j)]$$

- if $H_{t-2} \in \{N_r, \forall r \in V\}$ then O_{t-1} was emitted from the final network of r and a model transition occurs at t. Hence O_t will be emitted from the initial network of v with $H_t = 1_v$. In this case we need a maximization over n and r to specify the model that transits with the maximum likelihood. Therefore the likelihood term is computed as :

$$\delta_t^v(i,1_v,k) = a_{1_v k}^v b_{1_v k}^{Iv}(O_t) \max_{n,r} [\gamma_{t-1}^r(n,N_r) P(v|r)]$$

$P(v|r)$ is the transition probability from phone r to v and it is given by the language model obtained from frequency counts of phoneme pair occurrences.

Now in order to specify the value[4] of H_{t-2} that maximizes the overall likelihood for O_1^t we maximize among these two cases :

$$\delta_t^v(i,j,k) = \max \left\{ a_{ijk}^v b_{ijk}^v(O_t) \max_n [\delta_{t-1}^v(n,i,j)], \; a_{jk}^v b_{jk}^{Iv}(O_t) \max_{n,r} [\gamma_{t-1}^r(n,N_r) P(v|r)] \right\}$$

The case that yields the maximum likelihood specifies the value of H_{t-2} in the state sequence.

The complete algorithm is given as follows where we introduce $\psi_t(i,j)$ for back tracking the maximization arguments. Once the maximum likelihood is computed, the most likely state sequence is obtained by back tracking. the maximization arguments.

[4]Notice that, we allowed H_{t-2} to take values only in $\{1_v, \ldots, N_v, N_r, \forall r \in V\}$. For all other cases no state transitions are allowed and the likelihood is zero.

Initialization

$$\begin{aligned}\delta_1^v(j,k) &= a_j^v a_{jk}^v b_{jk}^{Iv}(O_1),\\ \gamma_1^v(j) &= 0,\\ \psi_1(j) &= 0\end{aligned}$$

Recursion for $1 < t < T$
$for\ 2_v \le j \le N_v$

$$\delta_t^v(i,j,k) = a_{ijk}^v b_{ijk}^v(O_t) \max_n[\delta_{t-1}^v(n,i,j)]$$
$$\psi_t(i,j) = \arg\max_{n,v}[\delta_{t-1}^v(n,i,j)]$$

$for\ j = 1_v$

$$\delta_t^v(i,j,k) = \max\left\{a_{ijk}^v b_{ijk}^v(O_t) \max_n[\delta_{t-1}^v(n,i,j)],\ a_{jk}^v b_{jk}^{Iv}(O_t) \max_{n,r}[\gamma_{t-1}^r(n,N_r)P(v|r)]\right\}$$

$$\psi_t(i,j) = \begin{cases} \arg\max_{n,v}[\delta_{t-1}^v(n,i,j)] \\ \arg\max_{n,v}[\gamma_{t-1}^r(n,N_r)P(v|r)]\end{cases}$$

$for\ j = N_v$

$$\gamma_t^v(i,j) = b_{ij}^{Fv}(O_t) \max_n[\delta_{t-1}^v(n,i,j)]$$

Termination

$$\begin{aligned}\gamma_T^v(i,j) &= b_{ij}^{Fv}(O_T) \max_n[\delta_{T-1}^v(n,i,j)]\\ \psi_T(i,j) &= \arg\max_{n,v}[\delta_{T-1}^v(n,i,j)]\\ P_{max}(O_1^T) &= \max_{i,j,v}[\gamma_T^v(i,j)]\\ (H_{T-1},H_T) &= \arg\max_{i,j}[\delta_T^v(i,j)]\end{aligned}$$

Back Tracking

$H_t = \psi_{t+2}(H_{t+1},H_{t+2}),\ t = T-2,\ldots,1.$

5.1 Complexity Analysis

Before describing the computational complexity of the proposed algorithm, we shall refer to our initial decoding algorithm. As briefly described before, the decoding is performed using Dawid's algorithm on a state augmented model (Deviren and Daoudi, 2002). The complexity of Dawid's algorithm depends on the size of each clique on the junction tree and the number of values of each variable in the clique. In our setting, the cliques[5] of the maximal network with structure $(\kappa^m, \tau_p^m, \tau_f^m) = (2,1,1)$ contains three consecutive hidden state variables (H_{t-1}, H_t, H_{t+1}). If we assume that $N_v = N\ \forall v \in V$, then each augmented variable takes $N|V|$ values. The number of distinct realizations for each clique is therefore $N^3|V|^3$. The computational complexity for an observation sequence of length T is $O(N^3|V|^3T)$.

For the algorithm proposed in this paper, there are three major items in the recursive computation. These are computed for all $t = 2,\ldots,T-1$ and $v \in V$. We analyze each of them separately for each t, v and then derive the asymptotic complexity accordingly.

- $a_{ijk}^v b_{ijk}^v(O_t) \max_n[\delta_{t-1}^v(n,i,j)]$
  This term is computed for $1 \le j \le N$. The complexity depends on the number of distinct values of the (i,j,k) triple. This would be N^3 without any topological constraints yielding $2N^3$ multiplications and N^3 operations in maximization. Exploiting the left-to-right topology for the state sequence, i.e. if $H_{t-1} = i$, then $H_t = j \in \{i, i+1\}$, the number of distinct values for (i,j,k) reduces to $4N+4$. The total number of operations is therefore (4N+4 + 2(4N+4)). Hence the complexity introduced by this term is $O(N)$ rather than $O(N^3)$.

- $a_{jk}^v b_{jk}^{Iv}(o_t) \max_{n,r}[\gamma_{t-1}^r(n,N)P(v|r)]$
  This term is computed only for $j = 1$. The computation requires $N+1$ multiplications for each k and $N|V|$ multiplications for each n,r. The maximization also introduces $N|V|$ operations. The asymptotic complexity is $O(N|V|)$. The topological constraint does not reduce the complexity because it only affects the range of values for k to reduce the $N+1$ multiplications to 3.

- $b_{ij}^{Fv}(O_t) \max_n[\delta_{t-1}^v(n,i,j)]$
  This term is computed only for $j = N$ with N multiplications and N^2 operations in maximization. The left-to-right topology limits the values of i to $\{N, N-1\}$. So

[5] A constraint triangulation scheme results in a repeating clique tree for the transition network. The cliques for the initial and final networks deviate from this repeating form. These are not considered for asymptotic complexity analysis.

the complexity reduces to $O(N)$ instead of $O(N^2)$.

Now considering the computation for all t and v, the overall asymptotic complexity without the topological constraints is $O(N^3|V|T + N^2|V|T + N|V|^2T)$. The improvement is significant as compared to $O(N^3|V|^3T)$. This improvement is due to 1) the parallel nature of the proposed algorithm and 2) exploitation of the structural properties of the DBNs we use. Further improvement is achieved by imposing the a left-to-right transition topology for the hidden states. In this case the dominant term in the asymptotic complexity is $O(N|V|^2T)$.

6 Experiments

In this section, we compare the performances of DBN models to standard HMMs. Our experiments are carried out on the TIMIT database. The TIMIT corpus of read speech has been designed to provide speech data for the acquisition of acoustic-phonetic knowledge and for the development and evaluation of automatic speech recognition systems (Lee and Hon, 1989). We present results where only the male part of the database with 2608 training sentences is used. The tests are performed on 128 sentences in the male core test set. In learning we train 48 phone models plus silence. Each phone is modeled with 3-states for HMMs and DBNs, i.e $H_t = 1, 2, 3$. The silence is modeled with a single state HMM. The output probability density of each observed variable is represented with a mixture of 8 Gaussians with diagonal covariance matrices. In tests, we use a bigram language model and neglect the confusions within 7 phone groups to yield 39 effective phones (Lee and Hon, 1989). The acoustic parameterization is based on standard Mel frequency cepstral coefficients (MFCC). The parameterization is performed on 25ms frames with a frame shift of 10ms. Cepstral coefficients are computed using a set of 24 Mel scaled triangular filters resulting in a vector of 11 static MFCCs (energy dropped). These are concatenated with first and second order derivatives to yield 35 features per frame.

In the first part of the experiment we constructed an HMM based system. The HMMs are learned using an embedded training procedure based on the phonetic transcriptions given in the database. We trained our models using the HTK toolkit. The phone accuracy [6] of this reference system on the male part of the core test set is **59.57** %. In the second part we use this system to obtain the best segmentation of each training sentence using a forced alignment procedure. Using this segmentation we run the structural learning algorithm to train DBN models for each phone. The upper bound on the structure search space is set to be $(2, 1, 1)$. The learned structure for each phone is given in Table 1. The maximal structure is $(2, 1, 1)$. The phone accuracy we obtain using our DBN system is **65.77** %. This shows that substantial gain in recognition accuracy can be obtained when model dependencies are learned from data. This supports our previous results on isolated and connected digit recognition experiments (Deviren and Daoudi, 2001) (Deviren and Daoudi, 2002). In addition, the computational complexity of decoding is greatly reduced using the proposed algorithm which enables the application of the methodology for complex speech recognition tasks.

(κ, τ_p, τ_f)	phones
(1,0,0)	b, g, dx, epi, zh, s#
(1,0,1)	dh, p, en, ch, hh
(1,1,0)	ng, el, y, uh, oy, th, d, v, aw
(1,1,1)	vcl, ah, ow, w, ix, ih, k, m, t, ey, ay, ae, er
(2,0,0)	
(2,0,1)	ax
(2,1,0)	sh, jh
(2,1,1)	z, l, s, cl, n, ao, r, q, aa, iy, f, eh, uw

Table 1: *Results of structural learning algorithm.*

[6] Better results on this database are readily achievable. Our goal here is not to tune the parameters in order to achieve the highest performances. Rather we want to provide a fair comparison between HMMs and DBNs using the same parameterization for both systems.

References

J. A. Bilmes. 2000. Dynamic Bayesian multinets. In *16th Conference on Uncertainty in Artificial Intelligence.*

O. Cetin, H. Nock, K. Kirchhoff, J. Bilmes, and M. Ostendorf. 2002. The 2001 GMTK-based SPINE ASR system. In *International Conference on Spoken Language Processing.*

K. Daoudi, D.Fohr, and C. Antoine. 2001a. Continuous multi-band speech recognition using Bayesian networks. In *Automatic Speech Recognition and Understanding Workshop.*

K. Daoudi, D.Fohr, and C. Antoine. 2001b. Dynamic Bayesian networks for multi-band automatic speech recognition. *to appear in Computer Speech and Language.*

A.P. Dawid. 1992. Application of a general propagation algorithm for probabilistic expert systems. *Statistics and Computing*, (2):25–36.

M. Deviren and K. Daoudi. 2001. Structural learning of dynamic Bayesian networks in speech recognition. In *7th European Conference on Speech Communication and Technology.*

M. Deviren and K. Daoudi. 2002. Continuous speech recognition using structural learning of dynamic Bayesian networks. In *11th European Signal Processing Conference.*

N. Friedman, K. Murphy, and S. Russell. 1998. Learning the structure of dynamic probabilistic networks. In *14th Conference on Uncertainty in Artificial Intelligence.*

U. Kjaerulf. 1992. A computational scheme for reasoning in dynamic probabilistic networks. In *8th Conference on Uncertainty in Artificial Intelligence*, pages 121–129.

Kai-Fu Lee and Hsiao-Wuen Hon. 1989. Speaker-independent phone recognition using hidden Markov models. *IEEE Trans. Acoustics, Speech and Signal Processing*, 37(11):1641–1648, November.

A. J. Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Trans. Information Theory*, 13:260–269.

G. Zweig, J. Bilmes, T. Richardson, K. Filali, K. Livescu, P. Xu, K. Jackson, Y. Brandman, E. Sandness, E. Holtz, J. Torres, and B. Byrne. 2002. Structurally discriminative graphical models for automatic speech recognition - results from the 2001 John Hopkins summer workshop. In *International Conference on Acoustics Speech and Signal Processing.*

G. Zweig. 1998. *Speech Recognition with Dynamic Bayesian Networks.* Ph.D. thesis, U.C., Berkeley.