Structural Learning of Dynamic Bayesian Networks in Speech Recognition

Murat Deviren, Khalid Daoudi

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

Abstract

We present a speech modeling methodology where no a priori assumption is made on the dependencies between the observed and the hidden speech processes. Rather, dependencies are learned from data. This methodology guarantees improvement in modeling fidelity compared to HMMs. In addition, it gives the user a control on the trade-off between modeling accuracy and model complexity. Furthermore, the approach is technically very attractive because all the computational effort is made in the training phase.

1. Introduction

First-order hidden Markov models (HMMs) are the mostly used statistical models in speech recognition. In HMMs, the observations are assumed to be governed by a hidden (unobserved) dynamic process, under some dependency assumptions. The latter state that the hidden process is first-order Markov, and that each observation depends only on the current hidden variable. There is however a fundamental question regarding these dependency assumptions: are they consistent with data? Then, if the answer is no: what (more plausible) dependency assumptions should we consider? However, we must keep in mind that HMMs lead to fast inference and learning algorithms. Therefore, any alternative dependency assumptions must also lead to fast recognition algorithms, otherwise they would be practically useless.

In this paper, we propose a methodology in which we do not make any a priori dependency assumptions. Rather, we give data a complete (but controlled) 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 the possible "realistic" dependencies, and to chose the ones which best explain the data. Our 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 make a trade-off between modeling accuracy and model complexity. In addition, the approach is technically very attractive because all the computational effort is made in the training phase.

Our approach is based on the framework of dynamic Bayesian networks (DBNs). The use of DBNs in speech recognition has recently gained attention [1, 2, 3, 4]. Briefly, the Bayesian networks (BNs) formalism consists in associating a directed acyclic graph to the joint probability distribution (JPD) \( P(X) \) of a set of random variables \( X = \{X_1, \ldots, X_n\} \). 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. The latter can be read as follows: conditioned on its parents, a variable is independent of all the other variables except its descendants. A BN is completely defined by a graph structure \( S \) and the numerical parameterization \( \Theta \) of the conditional probabilities of the variables given their parents. Indeed, the JPD can be expressed in a factored way as

\[
P(X) = \prod_{i=1}^{n} P(X_i | \Pi_i),
\]

where \( \Pi_i \) denotes the parents of \( X_i \). From this point of view, it is easy to represent an HMM as a DBN. Indeed, The Markov properties (the dependency assumptions) for an HMM are encoded by the graphical structure showed in Figure 1. Each node in this structure represents a random variable \( X_h[t] \) or \( X_o[t] \) whose configuration specifies the state or the observation at time \( t \).

Figure 1: HMM represented as a DBN

Instead of fixing a priori the structure (as is done with HMMs), we propose to build an "intelligent" system which works as follows. We feed the system using the observed data (the \( X_o[t] \)). Then, the system determines the structure \( S \) (i.e., the dependencies) and the parameters \( \Theta \) which best represent the data. This strategy is known as structural learning in the BNs literature. A related work has been proposed in [5], where a structural learning algorithm using Bayesian Multinets is performed. In [5], each element of the observation vector is considered as a separate variable, then time-cross dependencies among these variables are learned. The time window may include variables from the past and also from the future. The dependencies are chosen so as to reduce the entropy and to ensure that the new information is not already provided by the hidden state. Our work differs from [5] in two ways. First, we consider that short-time dependencies between the acoustic vectors are already represented by the \( \Delta \) and \( \Delta \Delta \) coefficients (we use the usual MFCC parameterization). In other words, we consider that direct dependencies between the observed vectors are not important. Second, our search strategy is different form the one of [5]. We use the structural EM algorithm proposed in [6] to find the optimal DBN.
In the next section, we define the class of structures (i.e., dependencies) we are interested in. In section 3, we first define the type of DBNs considered in our setting. We then describe briefly the structural learning algorithm. In section 4, we give the main lines of the learning algorithm when applied to the particular class of DBNs we consider. In section 5, we illustrate the performance of our approach on an isolated digit recognition task.

2. Search class

Searching over all the possible DBN structures would be computationally infeasible. In this section, we propose to reduce this search to only "realistic" dependencies, in a physical and computational sense.

First, we do not consider direct dependencies between observable variables. We believe this is a reasonable assumption because, short-time correlations between cepstral coefficients are indirectly considered by $\Delta$ and $\Delta\Delta$ coefficients. Second, we do not allow future dependencies between the hidden variables. This is because past dependencies must be considered given causality of speech. Thus, given that the graph has to be acyclic, we cannot have both past and future dependencies between hidden variables. Finally, we do not allow a hidden variable to have an observable one as a parent. This is because there exists no exact algorithm to infer BNs where continuous variables have discrete children. We also assume that the hidden process is stationary. Therefore, the remaining authorized dependencies in the class of structures we are considering are the following:

Let $X[t] = \{X_s[t], X_o[t]\}$ denote the hidden and observed variables at time $t$ respectively, and $X[t] = \{X[t], \ldots, X[T]\}$ represent the set of the variables for a sequence of length $T$.

- A hidden variable at time $t$ is independent of $X_{t-\kappa}^{t-1}$ given the last $\kappa$ hidden variables, if $t > \kappa$

$$P(X_s[t]|X_{t-\kappa}^{t-1}) = P(X_s[t]|X_{s}[t-\kappa], \ldots, X_s[t-1])$$ (2)

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

$$P(X_o[t]|X[t] \setminus \{X_o[t]\}) = P(X_o[t]|X_s[t-\tau_o], \ldots, X_s[t+\tau_i])$$ (3)

![DBN structure with $(\kappa, \tau_o, \tau_i) = (2, 1, 1)$, $T = 4$](image-url)

The stationarity assumption results in a repeating structure. Therefore, the triple $(\kappa, \tau_o, \tau_i)$ determines completely the DBN structure. When $(\kappa, \tau_o, \tau_i) = (1, 0, 0)$, the model reduces to the standard first-order HMM (Figure 1) where Eq.(2) defines the state transition probabilities and Eq.(3) defines the observation probabilities. When $(\kappa, \tau_o, \tau_i) = (2, 1, 1)$, we have the structure of Figure 2. Hence, the search class of possible DBN structures is defined by the triples $(\kappa, \tau_o, \tau_i)$, for $1 \leq \kappa \leq \kappa_{max}$, $0 \leq \tau_o \leq \tau_{max}$, $0 \leq \tau_i \leq \tau_{max}$, $(\kappa_{max}, \tau_{max}, \tau_{max})$ is an upper limit which defines the size of the search class.

In order to handle this problem we first show how DBNs can be used to model temporal processes with arbitrary dependency structure in time.

3. Learning dynamic Bayesian networks

3.1. Model definition

A DBN encodes the joint probability distribution of a time-evolving set $X[t] = \{X_1[t], \ldots, X_o[t]\}$ 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 joint probability density of $X[T] = \{X[1], \ldots, X[T]\}$ can be written as:

$$P(X[0], \ldots, X[T]) = \prod_{t=1}^{T} P(X[t]|\Pi_d)$$ (4)

where $\Pi_d$ denotes the parents of $X[t]$.

In the BNs literature, DBNs are defined using the assumption that $X[t]$ is Markovian [7]. In this paper, we consider processes that have dependency both to the past and to the future. More formally, we consider that the process $X[t]$ satisfies:

$$P(X[t]|X[t]_{t-\tau_o}^{t+\tau_i}) = P(X[t]|X[t-\tau_o], \ldots, X[t+\tau_i])$$ (5)

for some integers $\tau_o$ and $\tau_i$. Graphically, the above assumption states that a variable at time $t$ can only have parents in the interval $[t - \tau_o, t + \tau_i]$. However, care must be taken when dealing with boundary variables. Namely, for each of the first $\tau_o$ and the last $\tau_i$ time slices the dependency time window is different. This means that we do not have the repeating transition structure in these time slices. Therefore, the overall DBN structure is decomposed with respect to three time intervals: In $[1, \tau_o)$ and $[T - \tau_i, T]$, we have $\tau_o$ and $\tau_i$ different structures, respectively. In $[\tau_o + 1, T - \tau_i]$ the structure is repeating. Hence, the DBN representation simplifies to those with static BNs: $\tau_o$ initial networks, $\tau_i$ final networks and a transition network. This extends the definition used in [7], where the first order Markovianity assumption is made, resulting in one initial network and a transition network.

3.2. Structural learning

Learning the structure of a BN requires a scoring metric for each possible structure in the search class, and a search algorithm over the possible structures. The scoring metric is generally based on two terms, one maximizing the likelihood of the observed data, and one penalizing complex structures. The requirement for the penalty term is due to the fact that, as the structure complexity increases the likelihood of the data also increases in general. So in general, the structure that gives the
maximum likelihood is the one where all the variables are con-
nected to each other. However, such a model neglects the inde-
pendences among the variables, which causes the advantages of
BNs to be discarded. There are basically two approaches for pen-
zalizing complex structures. In the first one, a priori probability
distribution on structures is used. This scoring metric, known as
the Bayesian Dirichlet (BD) metric, gives high probabilities to
more realistic structures. The second approach, which is the one
we use, is known as the Bayesian Information Criterion (BIC)
(or Minimum Description Length (MDL)). The BIC score de-
fines a penalty term proportional to the number of parameters
used to encode data:

\[
\text{Score}_{BIC} = \log P(\Theta, S) - \frac{\log N}{2} \sum_{i=1}^{n} \|X_i, \Pi_i\| \tag{6}
\]

where \(D\) is the data set, \(N\) is the number of examples (realiza-
tions) in \(D\) and \(\|X, Y\|\) is defined as the number of parameters
required to encode the conditional probability, \(P(X|Y)\).

The evaluation of the scoring metric for all possible struc-
tures is generally not feasible. Therefore, many algorithms have
been proposed to search the structure space so as to achieve a
maximum scoring structure. In [6], a structural EM algorithm is
proposed to find the optimal structure and parameters for a BN,
in the case of incomplete data. The algorithm starts with ran-
dom structure and parameters. At each step, first a parametric
EM algorithm is performed to find the optimal parameters for
the current structure. Second, a structural search is performed to
increase the scoring. The evaluation of the scoring metric for
the next possible structure is performed using the parameters of
the previous structure. This algorithm guarantees an increase
in the scoring metric at each iteration, and converges to a local
maximum. We are going to use this algorithm in learning the
class of DBNs we are considering.

4. Learning within our search class

In this section, we present the main lines of the learning algo-
rithm for the class of DBN structures that we defined in section
2. We assume that each discrete hidden variable takes \(M\) val-
ues and each observable variable has a Gaussian density which
depend on the values of its parents. Therefore the numerical
parameterization of our DBNs is the following:

\[
P(X_h[t] \neq j | \Pi_h[t] = i) = a_{ij}[t], \quad \text{for } j = 1 \ldots M.
\]

\[
P(X_o[t] | \Pi_o[t] = i) \sim N(\mu_{i}, \Sigma_{i}) \tag{7}
\]

The (possibly) vector-index \(\iota\) is over all possible values of
the variable’s parents, which depend on the structure, i.e., on
the triple \((\kappa, \tau_p, \tau_l)\). Given the stationarity assumption, the
parameters of these conditional probabilities are constant over
time, when the structure is self-repeating, i.e.,

\[
a_{ij}[t] = a_{ij}, \quad \text{for } t = \kappa + 1, \ldots, T
\]

\[
\mu_{i}[t] = \mu_{i},
\]

\[
\Sigma_{i}[t] = \Sigma_{i}, \quad \text{for } t = \tau_p + 1, \ldots, T - \tau_l \tag{8}
\]

Given a fixed structure in our search class, we have to es-
imate the parameters relative to this structure which achieve
the maximum likelihood of data. This is done using the clas-
sical EM algorithm. However, the maximization of the auxiliary
function should be handled for each time \(t\) in the initial and fi-
nal networks. The update equations for a set of \(L\) observation
sequences \(O_t = \{o_{t}^{1}, \ldots, o_{t}^{L}\}\), each of length \(T_t\), are obtained as follows:

\[
\xi(t, j, i, t) = P(X_h[t] = j | \Pi_h[t] = i | O_t)
\]

\[
\gamma(t, i, t) = P(\Pi_h[t] = i | O_t) \tag{9}
\]

\[
\text{for } \kappa + 1 \leq t \leq T_t
\]

\[
a_{ij}[t] = \frac{\sum_{l=1}^{L} \sum_{\tau_{p}+1}^{\tau_{l}} \xi(t, j, i, t)}{\sum_{l=1}^{L} \sum_{j=1}^{M} \sum_{j=1}^{M} \xi(t, j, i, t)} \tag{10}
\]

\[
\text{for } 1 \leq t \leq \kappa
\]

\[
\mu_{i}[t] = \frac{\sum_{l=1}^{L} \sum_{\tau_{p}+1}^{\tau_{l}} o_{t}^{l} \gamma(t, i, t)}{\sum_{l=1}^{L} \sum_{i=1}^{M} \gamma(t, i, t)} - \mu_{i}[t] \tag{12}
\]

\[
\text{for } 1 \leq t \leq \tau_{p} \text{ and } T_t - \tau_{l} + 1 \leq t \leq T_t
\]

\[
\Sigma_{i}[t] = \frac{\sum_{l=1}^{L} \sum_{\tau_{p}+1}^{\tau_{l}} o_{t}^{l} \gamma(t, i, t)}{\sum_{l=1}^{L} \sum_{i=1}^{M} \gamma(t, i, t)} - \mu_{i}[t]^{T} \mu_{i}[t] \tag{13}
\]

The computation of \(\xi(t, j, i, t)\) and \(\gamma(t, i, t)\) is an inference
problem for BNs. We use the ILO algorithm [8], which allows
efficient computation of these quantities.

The learning algorithm can now be stated as follows:

- Specify the triple \((\kappa_{max}, \tau_{max}, \tau_{max})\), in order to de-
fine the search class.

- Run the structural EM algorithm. However, instead of
using a random structure in the first iteration, we use the
HMM structure. This way, we guaranty that the learned
model will have a likelihood greater (or equal) than the
HMM likelihood. In other words, the learned model is
guaranteed to model speech with higher fidelity than
HMMs.

The triple \((\kappa_{max}, \tau_{max}, \tau_{max})\), controls the size of
the set of allowed structures. Therefore, a trade off can be made
at this level over the complexity of the learning algorithm and the
fidelity of the resulting model.

An important point to note is that the structural search and
parameter updating for different structures is performed in the
training phase only. Once a specific model is learned for the
given data set, the recognition is performed using an inference
algorithm on the learned structure. In other words, all the com-
putational effort is done during training: which is a major tech-
nical advantage. Thus, the only measure of the computational
cost of our approach is given by the complexity of the inference
algorithm used in decoding.
5. Experiments

In this section, we provide a preliminary evaluation of the performance of our approach. Our primary goal here is to illustrate the potential of our approach. We are not trying to tune the parameters in order to achieve the best performances. The experiments are carried out on the isolated part of the Tidigits database in which 112 (resp. 113) speakers are used for training (resp. test). Each speaker uttered 11 digits twice. The size of each observation vector is 35, consisting of 11 MFCC (energy dropped), 12 $\Delta$, and 12 $\Delta\Delta$. Each hidden variable takes 4 values, $M = 4$. Each observable variable has a single Gaussian distribution with a diagonal covariance matrix. We assume a left-to-right transition topology for all the structures considered.

For the purpose of this experiment the upper bound on the dependence parameters are chosen to be $(\kappa_{mnp, \tau_{mn}}, \tau_{mn}, \tau_{mf_{mn}}) = (2, 1, 1)$. We applied the structural learning algorithm using BIC scoring for each digit. The algorithm converges to the same structure for all digits with dependency parameters as, $(\kappa, \tau_{}, \tau_{}) = (1, 0, 1)$ (see Figure 3). This shows that future dependencies can be very important in modeling. The recognition rate obtained using this structure is 96.43%, while the rate given by the HMM structure is 92.86%. The improvement is remarkable given that the inference complexity of both structures is asymptotically the same. Indeed, due to the left-to-right topology assumption, the increase in the decoding complexity is only by a factor of 2 with respect to HMM decoding.

It is interesting to compare the recognition rates of all the structures within the search space. We performed separate parameter learning algorithms for each structure. In Table 1 we give a list of the recognition rates for each structure. The structures are listed in increasing structural complexity, where $(1,0,1)$-$\{1,1,0\}$ and $(2,0,1)$-$\{2,1,0\}$ have the same complexities respectively. One can see that, increasing model complexity generally increases the recognition performance. It should also be noted that none of the structures has a lower performance than the baseline HMM. BIC scoring leads to a good trade off between modeling fidelity and complexity. However we should observe that, although the structures $(1,0,1)$ and $(1,1,0)$ have the same complexity, and that $(1,0,1)$ has a higher likelihood for the training set, the recognition performance is higher for $(1,1,0)$. We relate this inconsistency to the imperfections in the data set.

<table>
<thead>
<tr>
<th>$\kappa$</th>
<th>$\tau_0$</th>
<th>$\tau_i$</th>
<th>Recognition Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>92.86</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>92.86</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>96.43</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>97.44</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>96.51</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>97.56</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>97.05</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>98.42</td>
</tr>
</tbody>
</table>

Table 1: Recognition rate for different model structures (%).

6. Conclusion

Using the DBNs framework, we developed an approach where the dependencies between the hidden and observed speech processes are learned from data. This approach has the advantage to guaranty an improvement in the modeling fidelity (compared to HMMs), while a control over the processing complexity is provided. In addition our approach is technically attractive since all the computational effort is done during the training phase.

7. References