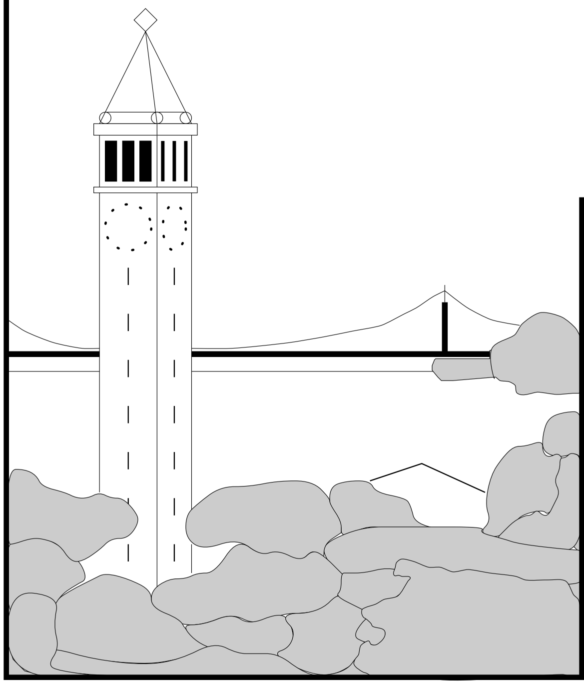
Dynamic Bayesian Networks and the Concatenation Problem in Speech Recognition

Geoffrey Zweig



Report No. UCB/CSD-96-927

December 1996

Computer Science Division (EECS)
University of California
Berkeley, California 94720

Dynamic Bayesian Networks and the Concatenation Problem in Speech Recognition

Geoffrey Zweig *

December 11, 1996

Abstract

This report describes a method for structuring dynamic Bayesian networks so that word and sentence-level models can be constructed from low-level phonetic models. This ability is a fundamental prerequisite for large-scale speech recognition systems, and is well-addressed in the context of hidden Markov models. With dynamic Bayesian networks, however, subword units cannot simply be concatenated together, and an entirely different approach is necessary.

1 Introduction

In the speech recognition task we are given a set of utterances, each of which has an associated word-level transcription. In the simplest case, each utterance is a single word, while in the most complex case the utterances consist of multiple words without intervening pauses. Based on this data, we want to construct probabilistic models of word pronunciation that will allow us to calculate the probability of any particular word sequence, given a sequence of acoustic observations.

Two important problems immediately arise: first, in speaker-independent systems meant to recognize thousands of words, there is rarely enough data to construct accurate models on a word-by-word basis. Hence it is necessary to learn models for a smaller number of atomic subword units, such as phonemes, and to combine these units as needed into larger structures. Secondly, while word-level transcriptions of the training utterances are available, they are not segmented into words, phonemes, or syllables. Hence, all reasonable partitionings of an utterance must be considered.

Hidden Markov models are currently the standard method for constructing probabilistic word models. A HMM assumes that an observation sequence $\{o_1, o_2..o_n\}$ is generated by a stochastic process that proceeds through a sequence of states $\{s_1, s_2..s_n\}$ and emits an observation symbol at each state. Thus the model is parameterized by a state transition matrix, and a matrix of emission probabilities. Thorough reviews of HMMs can be found in [2], [3], [4]. HMMs have proven their value in many practical applications; nevertheless, in their original formulation they have some serious drawbacks:

- They assume that the probability of an emission is conditionally independent of previous emissions; i.e. that it depends only on the current state. This assumption is violated by the fact that the observation symbols are typically generated from overlapping speech segments, and furthermore, because it is

*Research supported by NSF PYI Award IRI-9058426.

common to use both acoustic feature vectors *and their derivatives* as observations at each point in time.

- The states of a HMM do not necessarily correspond to the physical states of the actual generating process.
- It is not possible to capture the notion that the overall state of the system consists of a combination of separately describable factors.

The last two points have recently been addressed in [6], which describes a HMM variant with a factored state representation and an explicit physical interpretation. The Bayesian network formalism, however, solves these problems without ad-hoc modification, and this fact motivates our use of DBNs. The main advantage of these networks is that they work directly with the underlying variables of a physical system. In the case of speech recognition, these include the tongue position, the degree of lip-rounding, and other similar features. Furthermore, the overall state of the system is factored into a combination of these components, thus allowing for a compact representation. Finally, arbitrary conditional dependencies between the variables can be modeled in the most parsimonious way possible. Hence DBNs potentially offer a direct and efficient model of the speech generation process. A similar argument is presented in [7].

In order to exploit these advantages, however, it must be possible to combine atomic phonetic models into word and sentence-level structures. The remainder of this paper describes a method of doing this, and thus allows for a practical implementation.

2 DBN Formalism

A Bayesian network consists of two things [8]:

1. A collection of random variables. The values of some of these variables are known (observable variables), and the values of others are unknown (hidden variables). Each variable is represented by a node in the network.
2. A specification of the joint probability distribution of the variables. This is specified with the aid of the chain rule of probabilities. First the variables are assigned an ordering; then the parents of each variable are identified. The parents of a variable X_i must occur before X_i in the ordering, and given values for its parents, X_i is conditionally independent of all other variables occurring before it in the ordering. The parents of variable X_i are conventionally referred to as Π_i. Hence the probability of any particular assignment of values to the variables can be calculated as:
$P(x_1, x_2, ...x_n) = \prod_i P(x_i|\pi_i)$
The necessary conditional probabilities are enumerated in conditional probability tables (CPTs) associated with the variables.

The key advantage of a Bayesian network over a full joint probability table is that it more efficiently represents the same information. When each variable is conditionally dependent on only a small number of parents, an exponential reduction in space can be achieved.

A dynamic Bayesian network is a segmented Bayesian network in which each segment represents the collection of variables at a specific point in time. As defined, a dynamic Bayesian network is extremely general, and as a special case it reduces to a HMM. The restrictions necessary for this reduction are:

1. The nodes in each segment or timeslice are further subdivided into "observable" and "state" nodes. Values for observable nodes are always available, while values for state nodes are never available.

2. Arcs can only connect:
   - state nodes to observation nodes in the same timeslice or
   - observation nodes within a timeslice or
   - state nodes within a timeslice or
   - state nodes in timeslice t to state nodes in timeslice $t+1$.

Given a DBN, inference procedures can be used to calculate

1. the probability of observed data,
2. the likeliest instantiation of the hidden nodes given observed data,
3. the marginal distributions of the hidden nodes given observed data, and
4. conditional probability tables for the hidden nodes, via an expectation-maximization (EM) procedure.

Because a DBN is a special kind of Bayesian network, the inference methods described in [8], [9], [10] can all be used. The EM procedure is described in [5].

3 Word Models, Subword Models, and the Concatenation Problem

The training data consists of a set of utterances $\{U_1, U_2..U_n\}$, each with an associated word-level transcription: $\{\{w_{11}, w_{12}..w_{1i}\}, \{w_{21}, w_{22}..w_{2j}\}..\{w_{n1}, w_{n2}..w_{nk}\}\}$. With the aid of a dictionary, the word-level transcriptions can be turned into phonetic transcriptions: $\{\{p_{11}, p_{12}..p_{1q}\}, \{p_{21}, p_{22}..p_{2r}\}..\{p_{n1}, p_{n2}..p_{ns}\}\}$. With both HMMs and DBNs, we want to learn a probabilistic model for each phoneme, and then to construct word and sentence-level models through composition. (Section 3.1.1 shows how to accommodate multiple phonetic models for each word.)

Figure 1 gives an example of this process with HMMs. The top half of Fig. 1 shows models for the phonemes "n" and "o" separately. There are initial and final states, which are dummy states, and three intermediate states. Each of the intermediate states has associated emission probabilities. The bottom half of Fig. 1 shows a composite model for the word "no". Note that subword models can be concatenated without explicitly partitioning the observation sequence between the different model components. The sequence of allowable state transitions which is specified by a HMM does not make any explicit reference to time; it is *time-indefinite*.

Unfortunately, the semantics of DBNs arcs prevent an analogous concatenation process: arcs represent conditional dependencies, not allowable state transitions. DBN are *time-definite*: each segment refers to a specific point in time, and the concatenation of phoneme-specific segments would require an explicit partitioning of the observation sequence. Since this partitioning is unknown, a fundamentally different approach is needed, and in the following sections we present a solution to this problem.

3.1 DBN Control Structures

Figure 2 shows a very simple generative model in the DBN context. Each index node specifies the index of the phoneme being emitted by the timeslice it occurs in (is it the first phoneme of the utterance, the second,

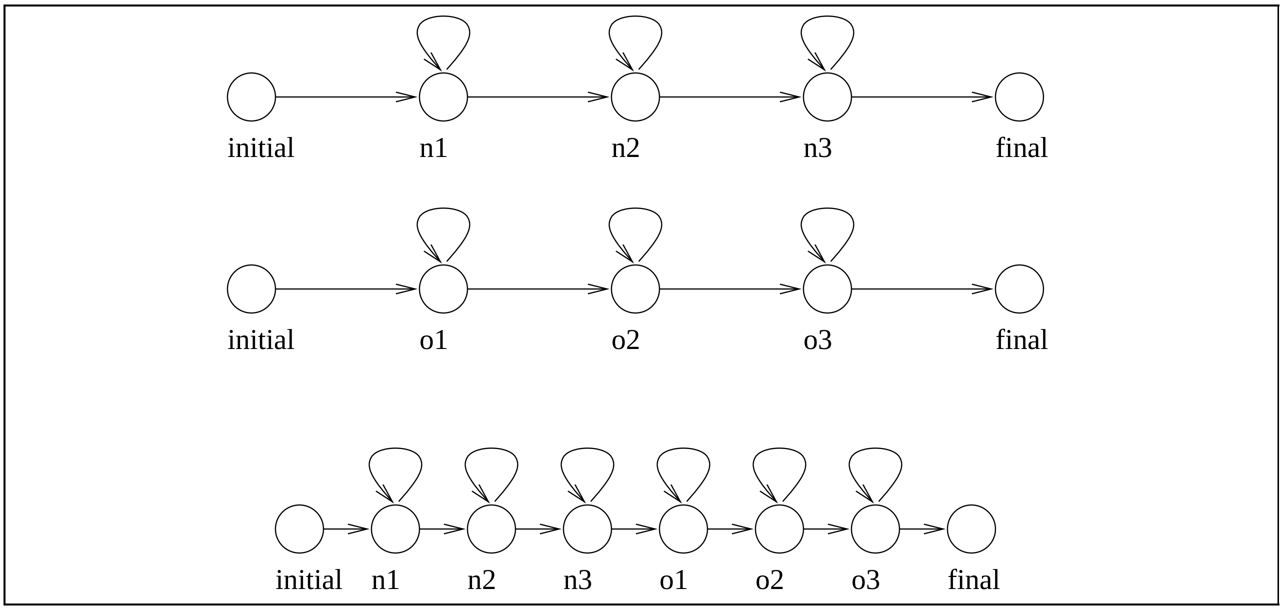


Figure 1: HMM model concatenation.

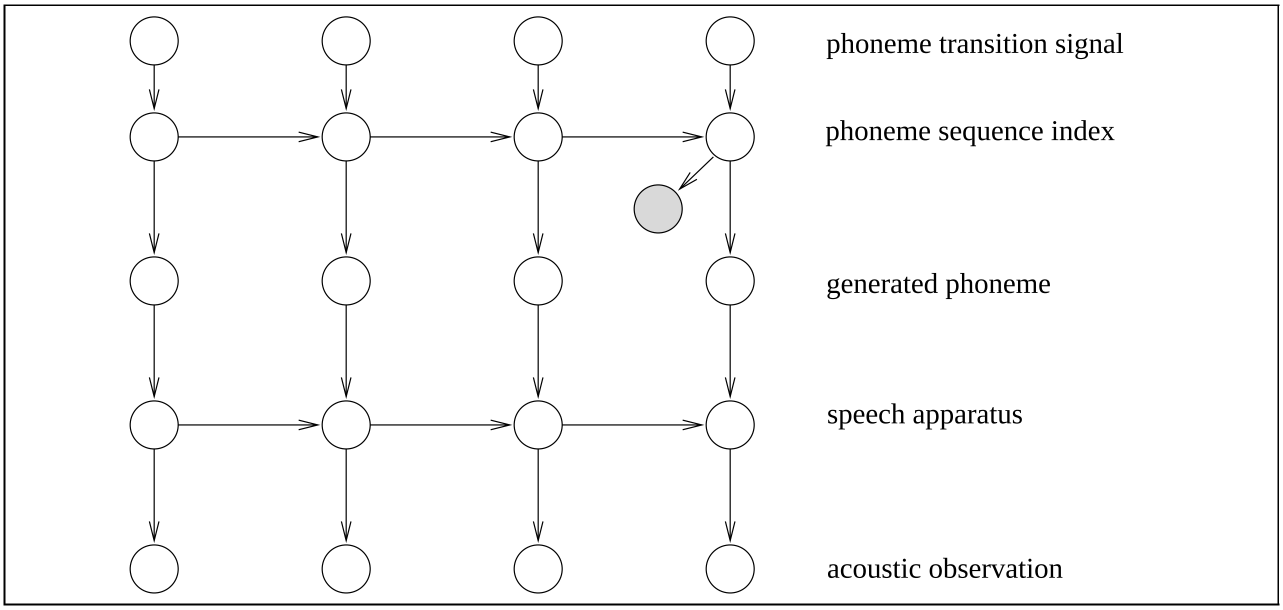


Figure 2: A simple generative model. Elaborations might include the addition of arcs between observation nodes to model conditional dependencies, or the addition of arcs from phoneme nodes in one timeslice to transition nodes in the next. The final dummy observation node is shaded.

Node Type	Deterministic CPTs	Example-Specific CPTs	Learned CPTs
Transition	N	N	N
Index	Y	Y	N
Phoneme	Y	Y	N
Speech App.	N	N	Y
Acoustic Obs.	N	N	Y

Table 1: The properties of the different node types.

etc.). A mapping particular to each utterance maps the index to an actual phoneme, and this mapping is hard-coded into the CPTs of the generated-phoneme nodes. Hence these CPTs must be adjusted to reflect the transcription of each utterance. The transition nodes are binary, with the value 1 marking a transition between two phonemes.

The prior on the first index node ensures that the model begins with the first phoneme in the transcription. A dummy observation node is added to the last timeslice to ensure that the model ends with the last phoneme. This is done by "observing" the node to have a value that is only possible if its parent, the index node, has a value equal to the number of phonemes in the utterance. The CPTs of the index-node are deterministic and force the index value to increase by 1 every time the transition signal is 1. The transition nodes have prior probabilities arbitrarily set to the non-zero value of 0.5. Since the DBN inference process will consider all possible assignments of values to the hidden nodes, all possible partitionings of the observation sequence will be examined.

Each speech-apparatus node represents a sub-network of physically meaningful variables, for example tongue position or vocal cord activity. The values of these variables are dependent on their values at the previous time step - thus expressing continuity constraints - and on the phoneme being generated. Finally, the acoustic observations are dependent on the speech apparatus. All the nodes in this model are hidden, except for the acoustic observations. Only the speech-apparatus and acoustic-observation nodes have CPTs which are subject to training. Table 1 summarizes the properties of each node type.

We refer to the transition, index, and generated-phoneme nodes as the network control nodes. Although it is not immediately obvious, the CPTs of the control nodes do not introduce bias. Since every explanation of an example has the same number of time steps s, the factor contributed by the transition-nodes to the overall probability (i.e. 2^{-s}) is the same for all explanations. (By "explanation" we mean an assignment of values to the hidden nodes.) Furthermore, before the statistics from different examples are combined, the statistics for each individual example are divided by the probability assigned to the example (see [5]). Since the control nodes contribute the same factor to every explanation of an example, this factor is also present in the overall example probability, and the division eliminates it. In fact, the control nodes never have unwanted effects; in particular they have no ill-effects in these situations:

- In EM because of the division just mentioned;
- In finding the likeliest explanation of a single example because the factor contributed is the same for every explanation, and nullified by the maxing operation;
- In computing marginals again because the factor is the same for each explanation, and removed by normalization;
- And when comparing models because the factor is the same for each model of an utterance.

Finally, we note that in general the transition probabilities will be phoneme-dependent and learned with EM. In this case there is no bias because all the other control CPTs have entries which are either 0 or 1.

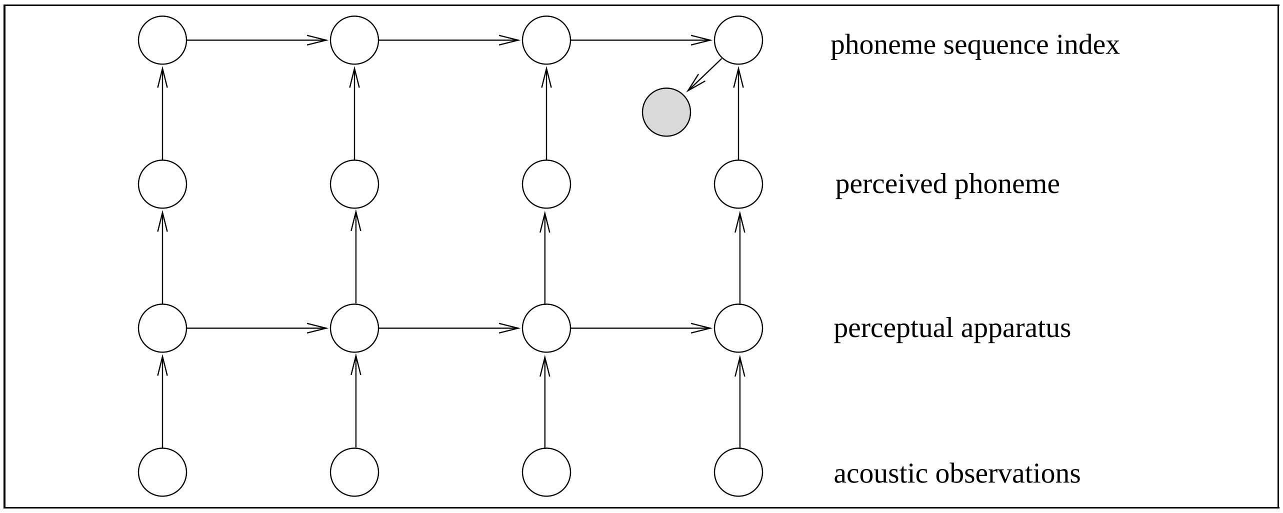


Figure 3: A perceptual model.

Hence the control nodes will have the effect of contributing a factor of 1.0 to all explanations that respect the phonetic transcription, and of rejecting all explanations that violate it.

3.1.1 Multiple Word Models

In the previous section we assumed that there is a single phonetic transcription for each word. Hence we were able to construct a unique phonetic transcription for the sequence of words in each utterance. A simple modification of the index-node CPTs allows for multiple word models. Here we think of the index-node as taking values C_{ijk} indicating that the timeslice is explaining the k^{th} phoneme of the j^{th} word model for the i^{th} word of the word sequence. Transitions are allowed sequentially within a word model, as before, and with equal probability between models when the end of a model is reached.

Since each explanation has the same number of word-transitions, and the number of possibilities faced at each transition is the same, the factor contributed by the control variables to the overall example probability is again the same for all explanations.

4 Extensions

4.1 A Perceptual Model

Figure 3 shows a simple perceptual model. In contrast to the generative model, information flows upwards from the acoustic observations. Here the values of the central variables - those associated with the perceptual apparatus - are causally determined by the observations.

As in the generative model, the index nodes ensure that the phonetic transcription is respected. If the perceived phoneme at time t is the same as the phoneme perceived at $t-1$, then the index remains unchanged. If the perceived phoneme at time t is consistent with the next phoneme in the transcription, then the index is incremented. Otherwise, the index takes a special "out-of-bounds" value which never changes. Since an explanation, after admitting an inconsistency, can never end up with correct final index value, it will

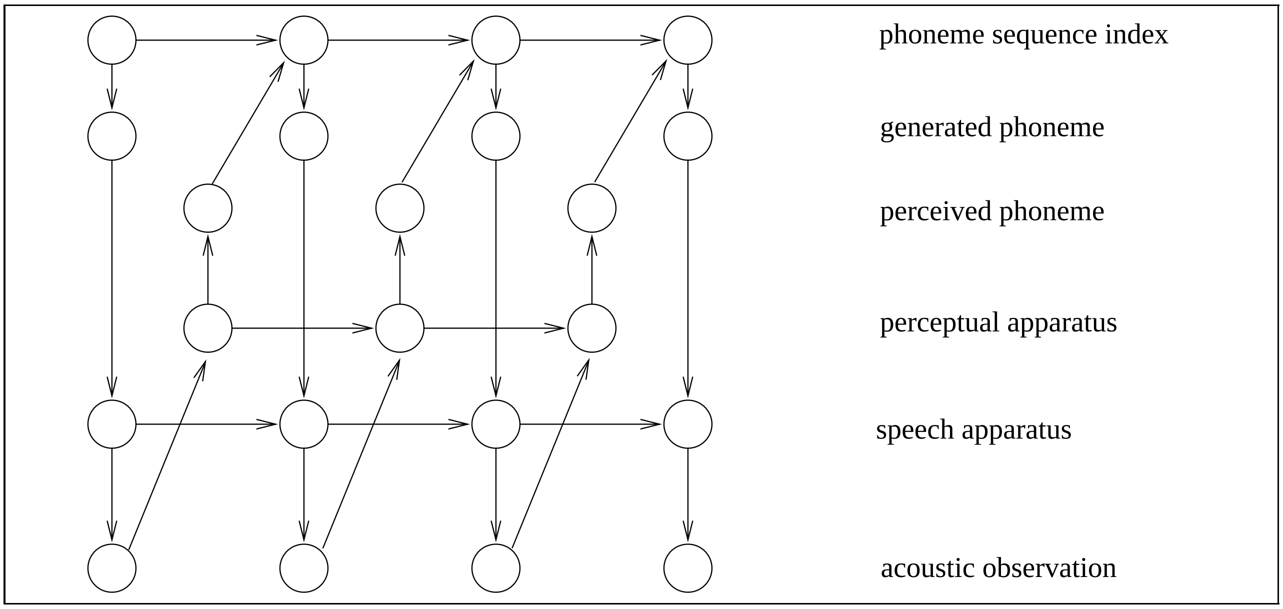


Figure 4: A combined generative and perceptual model.

be assigned zero probability by the final dummy observation and ignored. Here we have assumed that two occurrences of a phoneme do not occur consecutively; if desired, models of this sort can be made with explicit transition variables.

4.2 Combined Models

Figure 4 shows a combined generative and perceptual model. Note that an arc from the perceived phoneme to the index node in the same time-step would not make sense because perception occurs *after* generation. Hence the loop is completed with an arc from the perceived phoneme to the index node of the next time-step. The CPTs of the index nodes are again set so that the index value at time $t+1$ can either be the same as the previous time step, one-greater, or "out-of-bounds" in the case of an inconsistency between the generated and perceived phonemes. Explicit transition-signal nodes may be added. Different observations can be used for the generative and perceptual models, provided that appropriate arcs are introduced between the different kinds of evidence to reflect their interdependencies.

4.3 Noise Models

Figure 5 shows how a noise model can be expressed in this framework; it could be grafted onto any one of the other models.

5 Conclusions

Dynamic Bayesian networks provide a concise and flexible means of describing generative and perceptual processes. Due to their time-definite nature, however, the composition of low-level phonetic models into

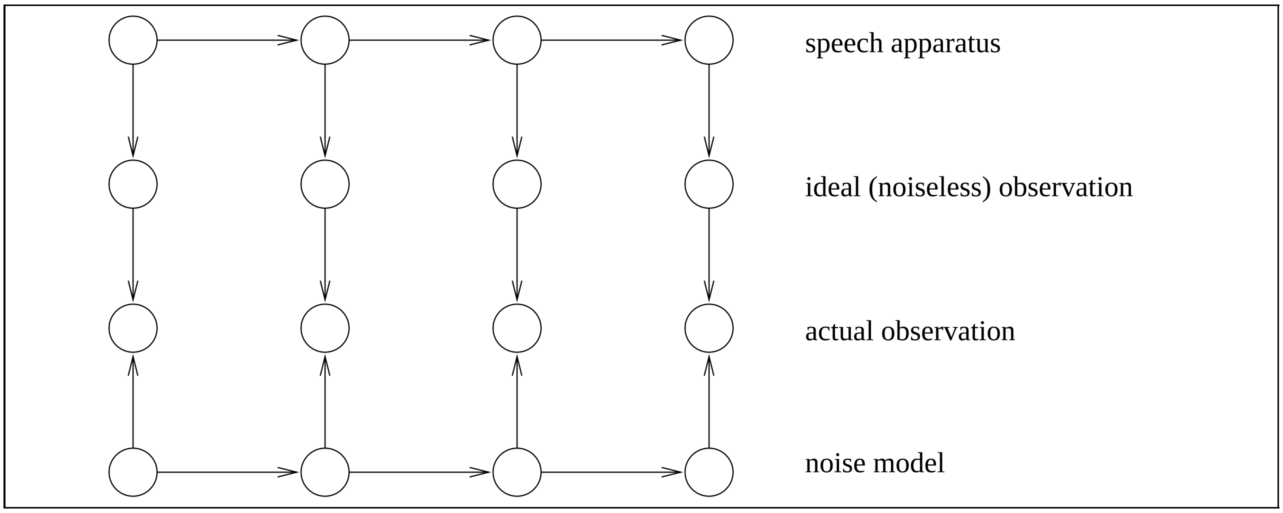


Figure 5: A noise model.

word and sentence-level models requires special techniques. The contribution of this report is a method of structuring DBNs that enables this sort of composition.

We are currently in the process of experimenting with detailed models of speech generation. As a proof-of-concept, without the speech-apparatus level, we have already achieved good results on a multi-speaker database of spoken digits. In the future we will extend this work to perceptual and combined models.

Acknowledgments I am grateful to Stuart Russell and Nelson Morgan for many useful conversations. This work further benefited from conversations with Nir Friedman, Paul Horton, and Kevin Murphy.

References

[1] R.M. Chavez and G.F. Cooper. A Randomized Approximation Algorithm for Probabilistic Inference on Bayesian Belief Networks. *Networks.* vol. 20. 661-685. 1990.

[2] L. Rabiner and B. Juang. *Fundamentals of Speech Recognition.* Prentice-Hall Signal Processing Series. 1993.

[3] K. Lee. *Automatic Speech Recognition. The Development of the SPHINX System.* Kluwer Academic Publishers. 1989.

[4] J.R. Deller, Jr., J.G. Proakis, and J.H.L. Hansen. *Discrete-Time Processing of Speech Signals.* Macmillan. 1993.

[5] D. Heckerman. A Tutorial on Learning Bayesian Networks. *Microsoft Research Technical Report* MSR-TR-95-06. 1995.

[6] L. Deng. Speech Recognition Using Autosegmental Representation of Phonological Units with Interface to the Trended HMM. *Free Speech Journal.* vol. 1. 1996.

[7] P. Smyth, D. Heckerman, M.I. Jordan. Probabilistic Independence Networks for Hidden Markov Probability Models. *Microsoft Research Technical Report.* MSR-TR-96-03. 1996.

[8] J. Pearl. *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann Publishers. 1988.

[9] J.F. Jensen, S.L. Lauritzen, K.G. Olesen. Bayesian Updating in Recursive Graphical Models by Local Computations. *Computational Statistical Quarterly.* v.4. pp.269-282. 1990.

[10] G. Zweig. A Forward-Backward Algorithm for Inference in Bayesian Networks and An Empirical Comparison with HMMs. Master's Thesis, University of California at Berkeley. 1996.