Speech Recognition with Dynamic Bayesian Networks

Geoffrey Zweig and Stuart Russell
Computer Science Division, UC Berkeley
Berkeley, California 94720
{zweig,russell}@cs.berkeley.edu

Abstract

Dynamic Bayesian networks (DBNs) are a useful tool for representing complex stochastic processes. Recent developments in inference and learning in DBNs allow their use in real-world applications. In this paper, we apply DBNs to the problem of speech recognition. The factored state representation enabled by DBNs allows us to explicitly represent long-term articulatory and acoustic context in addition to the phonetic-state information maintained by hidden Markov models (HMMs). Furthermore, it enables us to model the short-term correlations among multiple observation streams within single time-frames. Given a DBN structure capable of representing these long- and short-term correlations, we applied the EM algorithm to learn models with up to 500,000 parameters. The use of structured DBN models decreased the error rate by 12 to 29% on a large-vocabulary isolated-word recognition task, compared to a discrete HMM; it also improved significantly on other published results for the same task. This is the first successful application of DBNs to a large-scale speech recognition problem. Investigation of the learned models indicates that the hidden state variables are strongly correlated with acoustic properties of the speech signal.

Introduction

Over the last twenty years, probabilistic models have emerged as the method of choice for large-scale speech recognition tasks in two dominant forms: hidden Markov models (Rabiner & Juang 1993), and neural networks with explicitly probabilistic interpretations (Bourlard & Morgan 1994; Robinson & Fallside 1991). Despite numerous successes in both isolated-word recognition and continuous speech recognition, both methodologies suffer from important deficiencies. HMMs use a single state variable to encode all state information; typically, just the identity of the current phonetic unit. Neural networks occupy the opposite end of the spectrum, and use hundreds or thousands of hidden units that often have little or no intuitive meaning.

Our work is motivated by the desire to explore probabilistic models that are expressed in terms of a rich yet well-defined set of variables, and dynamic Bayesian networks provide the ideal framework for this task: with a single set of formulae expressed in a single program, probabilistic models over arbitrary sets of variables can be expressed and computationally tested. By decomposing the state information into a set of variables, DBNs require fewer parameters than HMMs to represent the same amount of information. In the context of speech modeling, DBNs provide a convenient method for defining models that maintain an explicit representation of the lips, tongue, jaw, and other speech articulators as they change over time. Such models can be expected to model the speech generation process more accurately than conventional systems. One particularly important consequence of including an articulatory model is that it can handle *coarticulation* effects. One of the main reasons these occur is that the inertia of the speech articulators which is acquired in the generation of one sound modifies the pronunciation of following sounds. In addition, DBNs are able to model the correlations among multiple acoustic features at a single point in time in a way that has not previously been exploited in discrete-observation HMMs.

We have implemented a general system for doing speech recognition in the Bayesian network framework, including methods for representing speech models, efficient inference methods for computing probabilities within these models, and efficient learning algorithms for training the DBN model parameters from observations. The system has been tested on a large-vocabulary isolated-word recognition task. We found that a large improvement results from modeling correlations among acoustic features within a single time frame. A further increase results from modeling the temporal correlations among acoustic features across time frames. Analysis of the learned parameters shows that the two kinds of models capture different aspects of the speech process.

Problem Background

The task of a statistical speech recognition system is to learn a parametric model from a large body of training data, and then to use the model to recognize the words in previously unheard utterances. Since the number of words in a natural language is large, it is impossible to learn a specific model for every word. Instead, words are expressed in terms of a small number of stereotypical atomic sounds or phonemes—English, for example, is often modeled in terms of 40 to 60 phonemes. Models for each phoneme are learned, and whole-word models are created by concatenating the models of the word's constituent phonemes. So, for example, the word "cat" might have the phonetic transcription */k ae t/*.

In order to model coarticulatory effects, expanded phonetic alphabets are often used, in which there is a unique symbol for each phoneme in the context of surrounding phonemes. In left-context biphone alphabets, there is a phonetic unit for each phoneme in the left-context of every possible preceding phoneme. In right-context biphone alphabets, there is a unit for each phoneme in the right-context of every possible following phoneme. Triphone modeling is a particularly common scheme in which there is a unit for each phoneme in the context of all possible preceding *and* following phonemes. The phonetic units found in these (and other) alphabets are often referred to as *phones*. Theoretically, the use of biphones squares the number of atomic units, and the use of triphones cubes the number; in practice, only the commonly occurring combinations are modeled.

It is often beneficial to break each phonetic unit into two or more substates. In a two-state-per-phone system, for example, each phone is broken into an initial sound and a final sound, thus doubling the total number of phonetic units.

Whatever the precise form of the phonetic alphabet, the training data consists of a collection of utterances, each of which has an associated phonetic transcription. Each utterance is broken into a sequence of overlapping time frames, and the sound is processed to generate the acoustic features $o_1, o_2, \ldots, o_n$. One or more acoustic features may be extracted from each frame, and we use the notation o_i to refer to the features extracted from the ith frame regardless of number. A phonetic transcription or word model, M, is also associated with each utterance.

Statistical Speech Recognition

The main goal of a statistical speech recognition system is to estimate the probability of a word model M given a sequence of acoustic observations $\mathbf{o}$. (We focus on isolated word recognition, and the results generalize to connected word recognition.) This can be rewritten with Bayes' rule as: $P(M|\mathbf{o}) = \frac{P(\mathbf{o}|M)P(M)}{P(\mathbf{o})}$. This is desirable because it decomposes the problem into two subproblems: $P(M)$ can be estimated from a language model that specifies the probability of the occurrence of different words, and $P(\mathbf{o}|M)$ can be estimated with a model that describes how sounds are generated. Since $P(\mathbf{o})$ is a constant with respect to word models, different models M_i can be compared by computing just $P(\mathbf{o}|M_i)P(M_i)$. Computation of $P(M)$ is straightforward in the case of isolated words, and we focus on the estimation of $P(\mathbf{o}|M)$, i.e., the probability of the observation sequence given the word.

This probability distribution is not usually estimated directly. Instead, statistical models typically use a collection of hidden state variables $\mathbf{s}$, which are intended to represent the state of the speech generation process over time. Thus we have

$$P(\mathbf{o}|M) = \sum_{\mathbf{s}} P(\mathbf{o}, \mathbf{s}|M)$$

In addition, the observation generated at any point is usually assumed to depend only on the state of the process, so we have

$$P(\mathbf{o}|M) = \sum_{\mathbf{s}} P(\mathbf{s}|M)P(\mathbf{o}|\mathbf{s})$$

We refer to the specification of $P(\mathbf{s}|M)$ as the pronunciation model, and to the specification of $P(\mathbf{o}|\mathbf{s})$ as the acoustic model.

HMMs

A hidden Markov Model is a simple representation of a stochastic process of the kind described above. The hidden state of the process is represented by a single state variable s_i at each point in time, and the observation is represented by an observation variable o_i (Figure 1). Furthermore, a Markovian assumption is made, so that we can decompose the probability over the state sequence as follows (leaving implicit the dependence on M):

$$P(\mathbf{o}, \mathbf{s}) = P(s_1)P(o_1|s_1)\prod_{i=2}^{n} P(s_i|s_{i-1})P(o_i|s_i)$$

In the case of speech, the state variable is usually identified with the *phonetic state*, i.e., the current phone being uttered. Thus, the pronunciation model is contained in the probability distribution $P(s_i|s_{i-1}, M)$ which designates the transition probabilities among phones, and consequently the distribution over phone sequences for a particular word. The acoustic model is the probability distribution $P(\mathbf{o}|\mathbf{s})$, and is independent of the particular word. Both of these models are assumed independent of time.

The conditional probability parameters in HMMs are usually estimated by maximizing the likelihood of the observations using the EM algorithm. Once trained, the HMM is used to recognize words by computing $P(\mathbf{o}|M_i)$ for each word model M_i. For details, the reader is referred to (Rabiner & Juang 1993).

In this paper, we will be concerned with *discrete* observation variables, which can be created from the actual signal by the process of *vector quantization* (Rabiner & Juang 1993). In order to allow for a wide range of sounds, it is common to generate several discrete observation variables o_i^j at each point in time, each of which has a fairly small range (say 256 values). To

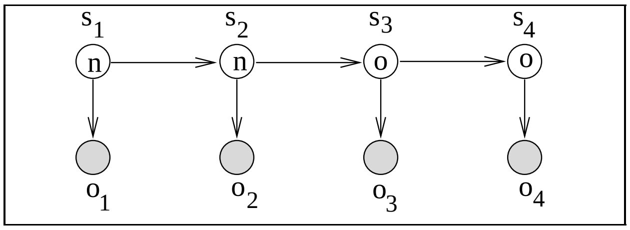

Figure 1: A DBN representation of an HMM. There is a distinct state and observation variable at each point in time. A node in the graph represents a variable, and the arcs leading into a node specify the variables on which it is conditionally dependent. A valid assignment of values to the state variables for the word "no" is shown. Observation variables are shaded. This simple picture ignores the issues of parameter tying and phonetic transcriptions.

keep the number of parameters manageable with these multiple observation streams, a further conditional independence assumption is typically made (Lee 1989):

$$P(o_i|s_i) = \prod_j P(o_i^j|s_i)$$

Bayesian Networks

A Bayesian network is a general way of representing joint probability distributions with the chain rule and conditional independence assumptions. The advantage of the Bayesian network framework over HMMs is that it allows for an arbitrary set of hidden variables $\mathbf{s}$, with arbitrary conditional independence assumptions. If the conditional independence assumptions result in a sparse network, this may result in an exponential decrease in the number of parameters required to represent a probability distribution. Often there is a concomitant decrease in the computational load (Smyth, Heckerman, & Jordan 1997; Ghahramani & Jordan 1997; Russell *et al.* 1995).

More precisely, a Bayesian network represents a probability distribution over a set of random variables $\mathcal{V} = V_1,..V_n$. The variables are connected by a directed acyclic graph whose arcs specify conditional independence among the variables, such that the joint distribution is given by

$$P(v_1, \ldots, v_n) = \prod_i P(v_i|Parents(V_i))$$

where $Parents(V_i)$ are the parents of V_i in the graph. The required conditional probabilities may be stored either in tabular form or with a functional representation. Figure 1 shows an HMM represented as a Bayesian network. Although tabular representations of conditional probabilities are particularly easy to work with, it is straightforward to model observation probabilities with mixtures of Gaussians, as is often done in HMM systems.

When the variables represent a temporal sequence and are thus ordered in time, the resulting Bayesian network is referred to as a dynamic Bayesian network (DBN) (Dean & Kanazawa 1989). These networks maintain values for a set of variables X_i at each point in time. X_{ij} represents the value of the ith variable at time j. These variables are partitioned into equivalence sets that share time-invariant conditional probabilities.

Bayesian Network Algorithms. As with HMMs, there are standard algorithms for computing with Bayesian networks. In our implementation, the probability of a set of observations is computed using an algorithm derived from (Peot & Shachter 1991). Conditional probabilities can be learned using gradient methods (Russell *et al.* 1995) or EM (Lauritzen 1995). We have adapted these algorithms for dynamic Bayesian networks, using special techniques to handle the deterministic variables that are a key feature of our speech models (see below). A full treatment of these algorithms can be found in (Zweig 1998).

DBNs and Speech Recognition

Like HMMs, our DBN speech models also decompose into a pronunciation model and an acoustic model. However, our acoustic model includes additional state variables that we will call "articulatory context" variables; the intent is that these may capture the state of the articulatory apparatus of the speaker, although this will not be the case in all of our models. These variables can depend on both the current phonetic state and the previous articulatory context. Mathematically, this can be expressed by partitioning the set of hidden variables into phonetic and articulatory subsets: $\mathcal{S} = \mathcal{Q} \cup \mathcal{A}$. Then, $P(\mathbf{o}, \mathbf{s}|M) = P(\mathbf{o}, \mathbf{q}, \mathbf{a}|M) = P(\mathbf{q}|M)P(\mathbf{o}, \mathbf{a}|\mathbf{q})$. The Bayesian network structure can be thought of as consisting of two layers: one that models $P(\mathbf{q}|M)$, and one that models $P(\mathbf{o}, \mathbf{a}|\mathbf{q})$. Figure 2 illustrates a DBN structured for speech recognition in this manner. In the following two sections, we discuss the pronunciation model and acoustic model in turn.

Pronunciation Model. In (Zweig & Russell 1997; Zweig 1998), it is shown that the DBN model structure we use can represent any distribution over phone sequences that can be represented by an HMM. For the purposes of simplifying the presentation in this paper, we will make two additional assumptions. The first is that each word model consists of a linear sequence of phonetic units; so, for example, "cat" is assumed always to be pronounced /*k ae t*/ without any variation in the phonetic units present or their order. The second assumption concerns the average durations of phones, and is that the probability that there is a transition between two consecutive phones q_1 and q_2 is given by a phone-dependent transition probability, t_{q_1}.

The index node in Figure 2 keeps track of the position in the phonetic transcription; all words go through the same sequence of values $1, 2, \ldots, k$ where k is the

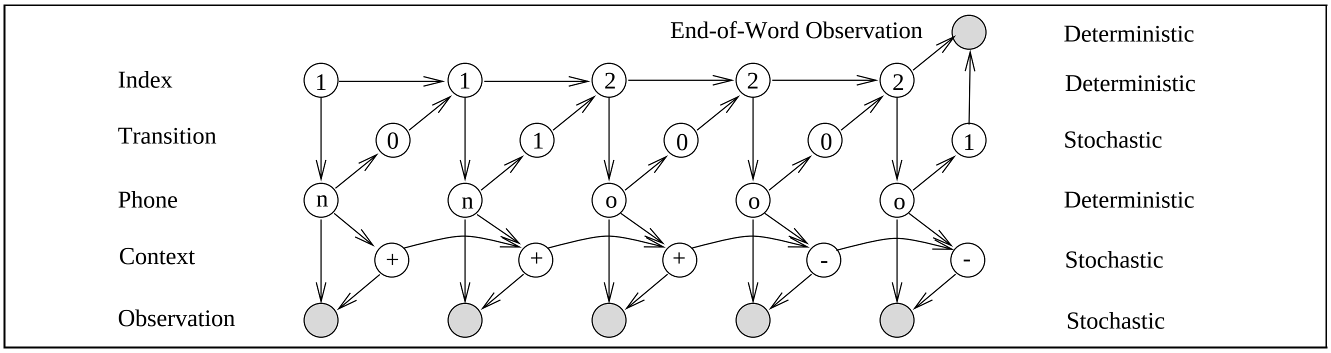

Figure 2: A DBN for speech recognition. The index, transition, phone, and end-of-word variables encode a probability distribution over phonetic sequences. The context and observation variables encode a distribution over observations, conditioned on phonetic sequence. A valid set of variable assignments is indicated for the word "no." In this picture, the context variable represents nasalization. The vowel */o/* is not usually nasalized, but in this case coarticulation causes nasalization of its first occurrence.

number of phonetic units in the transcription. An assignment of values to the index variables specifies a time-alignment of the phonetic transcription to the observations. For a specific pronunciation model, there is a deterministic mapping from the index of the phonetic unit to the actual phonetic value, which is represented by the phone variable. This mapping is specified on a word-by-word basis. There is a binary transition variable that is conditioned on the phonetic unit. When the transition value is 1, the index value increases by 1, which can be encoded with the appropriate conditional probabilities.

The distinction between phonetic index and phonetic value is required for parameter tying. For example, consider the word "digit" with the phonetic transcription */d ih jh ih t/*. The first */ih/* must be followed by */jh/*, and the second */ih/* must be followed by */t/*; thus there must be a distinction between the two phone occurrences. On the other hand, the probability distribution over acoustic emissions should be the same for the two occurrences; thus there should not be a distinction. It is impossible to satisfy these constraints with a single set of conditional probabilities that refers only to phonetic values or index values.

The conditional probabilities associated with the index variables are constrained so that the index value begins at 1 and then must either stay the same or increase by 1 at each time step. A dummy end-of-word observation is used to ensure that all sequences with non-zero probability end with a transition out of the last phonetic unit. This binary variable is "observed" to have value 1, and the conditional probabilities of this variable are adjusted so that $P(EOW = 1|index = last, transition = 1) = 1$, and the probability that $EOW = 1$ is 0 in all other cases. Conditioning on the transition variable ensures an unbiased distribution over durations for the last phonetic unit.

In Figure 2, deterministic variables are labeled. Taking advantage of the deterministic relationships is crucial for efficient inference.

Acoustic Model. The reason for using a DBN is that it allows the hidden state to be factored in an arbitrary way. This enables several approaches to acoustic modeling that are awkward with conventional HMMs. The simplest approach is to augment the phonetic state variable with one or more variables that represent articulatory-acoustic context. This is the structure shown in Figure 2.

The context variable serves two purposes, one dealing with long-term correlations among observations across time-frames, and the other with short-term correlations within a time-frame. The first purpose is to model variations in phonetic pronunciation due to coarticulatory effects. For example, if the context variable represents nasalization, it can capture the coarticulatory nasalization of vowels. Depending on the level of detail desired, multiple context variables can be used to represent different articulatory features. Model semantics can be enforced with statistical priors, or by training with data in which the articulator positions are known.

The second purpose is to model correlations among multiple vector-quantized observations within a single time-frame. While directly modeling the correlations requires a prohibitive number of parameters, an auxiliary variable can parsimoniously capture the most important effects.

Network Structures Tested. In our experiments, we tested networks that varied only in the acoustic model. All the DBN variants had a single binary context variable, and differed in the conditional independence assumptions made about this variable. We used the following model structures (see Figure 3):

1. An "articulator" network in which the context variable depends on both the phonetic state and its own past value. This can directly represent phone-dependent articulatory target positions and inertial constraints.

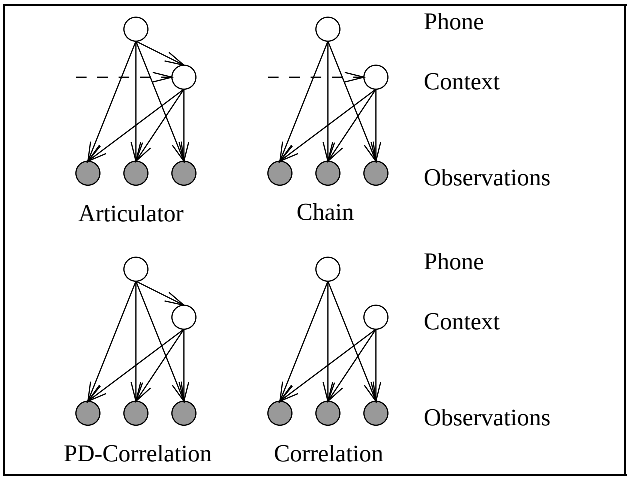

Figure 3: Acoustic models for the networks tested. Three acoustic observations were made in each time frame. The dotted arcs represent links to the previous time frame.

2. A "chain" network in which the phonetic dependence is removed. This structure can directly represent phone-independent temporal correlations.
3. A "phone-dependent-correlation" network (PD-Correlation) which results from removing the temporal links from the articulator network. This can directly model phone-dependent intra-frame correlations among multiple acoustic features.
4. A "correlation" network which further removes the phonetic dependence. This is only capable of modeling intra-frame observation correlations in the most basic way.

The articulator network was initialized to reflect voicing, and the chain network to reflect temporal continuity.

Experimental Results

Database and Task

As a test-bed, we selected the Phonebook database, a large-vocabulary, isolated-word database compiled by researchers at NYNEX (Pitrelli *et al.* 1995). The words were chosen with the goal of "incorporating all phonemes in as many segmental/stress contexts as are likely to produce coarticulatory variations, while also spanning a variety of talkers and telephone transmission characteristics." These characteristics make it a challenging data set.

The data was processed in 25ms windows to generate 10 mel-frequency cepstral coefficients (MFCCs) (Davis & Mermelstein 1980) and their derivatives every 8.4ms. MFCCs are generated by computing the power spectrum with an FFT; then the total energy in 20 different frequency ranges is computed. The cosine transform of the logarithm of the filterbank outputs is computed, and the low-order coefficients constitute the MFCCs. MFCCs represent the shape of the short-term power spectrum in a manner inspired by the human auditory system.

The MFCCs were vector-quantized using a size-256 codebook. Their derivatives were quantized in a second codebook. The C_0 and delta-C_0 coefficients were quantized separately with size-16 codebooks, and concatenated to form a third 256-valued data stream. We performed mean-cepstral subtraction for C_1 through C_{10}, and speaker normalization for C_0 (Lee 1989). The effect of mean-cepstral subtraction is to remove the transmission characteristics of telephone lines. Speaker normalization scales C_0 to the overall power level of a speaker by subtracting the maximum value, so that the resulting values can be compared across utterances.

We experimented with DBN models using both context-independent and context-dependent phonetic units. In both cases, we started from the phonetic transcriptions provided with Phonebook, ignoring the stressed/unstressed distinction for vowels.

In the case of context-independent units, i.e., simple phonemes, we used four distinct states for each phoneme: an initial and final state, and two interior states.

To generate the context-dependent transcriptions, we replaced each phoneme with two new phonetic units: one representing the beginning of the phoneme in the left-context of the preceding phoneme, and one representing the end of the phoneme in the right-context of the following unit. For example, the $/ae/$ in $/k\ ae\ t/$ becomes $/(k-\mathbf{ae})\ (\mathbf{ae}-t)/$. To prevent the proliferation of phonetic units, we did not use context-dependent units that were seen fewer than a threshold number of times in the training data. If a context-dependent unit was not available, we used a context-independent phoneme-initial or phoneme-final unit instead. Finally, we found it beneficial to repeat the occurrence of each unit twice. Thus, each phoneme in the original transcription was broken into a total of four substates, comparable to context-independent phonemes. The effect of doubling the number of occurrences of a phonetic unit is to increase the minimum and expected durations in that state.

We report results for two context-dependent phonetic alphabets: one in which units occurring at least 250 times in the training data were used, and one in which units occurring at least 125 times were used. In both cases, the alphabet also contained context-independent units for the initial and final segments of each of the original phonemes. The two alphabets contained 336 and 666 units respectively. Thus the number of parameters in the first case is comparable to the context-independent-alphabet system with an auxiliary variable; the number of parameters in the second case is comparable to the number that arises when an auxiliary variable is added to the first context-dependent system.

Note that the notion of context in the sense of a context-dependent alphabet is different from that rep-

Network	Parameters	Error Rate
Baseline-HMM	127k	4.8%
Correlation	254k	3.7%
PD-Correlation	254k	4.2%
Chain	254k	3.6%
Articulator	255k	3.4%

Figure 4: Test results with the basic phoneme alphabet; $\sigma \approx 0.25\%$. The number of independent parameters is shown to 3 significant figures; all the DBN variants have slightly different parameter counts.

resented by the context variable in Figures 2 and 3. Context of the kind expressed in an alphabet is based on an idealized pronunciation template; the context-variable represents context as manifested in a specific utterance.

The training subset consisted of all *a, *h, *m, *q, and *t files; we tuned the various schemes with a development set consisting of the *o and *y files. Test results are reported for the *d and *r files, which were not used in any of the training or tuning phases. The words in the Phonebook vocabulary are divided into 75-word subsets, and the recognition task consists of identifying each test word from among the 75 word models in its subset. There were $19,421$ training utterances, 7291 development utterances and 6598 test utterances. There was no overlap between training and test words or training and test speakers.

Performance

Figure 4 shows the word-error rates with the basic phoneme alphabet. The results for the DBNs clearly dominate the baseline HMM system. The articulatory network performs slightly better than the chain network, and the networks without time-continuity arcs perform at intermediate levels. However, most of the differences among the augmented networks are not statistically significant.

These results are significantly better than those reported elsewhere for state-of-the-art systems: Dupont *et al.* (1997) report an error rate of 4.1% for a hybrid neural-net HMM system with the same phonetic transcription and test set, and worse results for a more conventional HMM-based system. (They report improved performance with transcriptions based on a pronunciation dictionary from CMU.)

Figure 5 shows the word error rates with the context-dependent alphabets. Using a context-dependent alphabet proved to be an effective way of improving performance. For about the same number of parameters as the augmented context-independent phoneme network, performance was slightly better. However, augmenting the context-dependent alphabet with an auxiliary variable helped still further. We tested the best performing augmentation (the articulator structure) with the context-dependent alphabet, and obtained a significant performance increase. Increasing the alphabet size

Network	Parameters	Error Rate
CDA-HMM	257k	3.2%
CDA-Articulator	515k	2.7%
CDA-HMM	510k	3.1%

Figure 5: Test results with the context-dependent alphabets (CDA); $\sigma \approx 0.20\%$. In the first two systems, each context-dependent unit occurred at least 250 times in the training data; in the third, the threshold was 125. This resulted in alphabet sizes of 336 and 666 respectively.

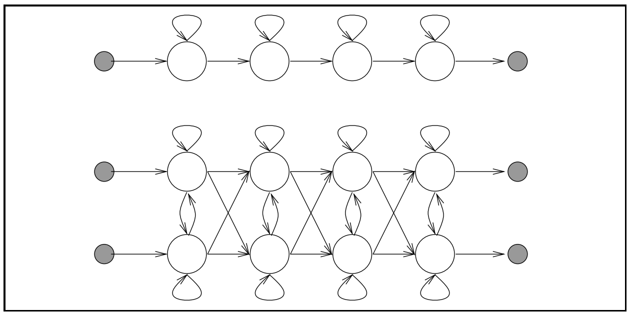

Figure 6: A 4-state HMM phone model (top), and a corresponding HMM model with a binary context distinction (bottom). In the second HMM, there are two states for each of the original states, representing context values of 0 and 1. The shaded nodes represent notional initial and final states in which no sound is emitted. Phone models are concatenated by merging the final state(s) of one with the initial state(s) of the other. The more complex model must have two initial and final states to retain memory of the context across phones. These graphs specify possible transitions between HMM states, and are not DBN specifications.

to attain a comparable number of parameters did not help as much.

In terms of computational requirements, the "Baseline-HMM" configuration requires 18M of RAM, and can process a single example through one EM iteration 6X faster than real time on a SPARC Ultra-30. The "Articulator" network requires 28M of RAM and runs 2X faster than real time.

Cross-Product HMM. Acoustic and articulatory context can be incorporated into an HMM framework by creating a distinct state for each possible combination of phonetic state and context, and modifying the pronunciation model appropriately. This is illustrated for a binary context distinction in Figure 6. In the expanded HMM, there are two new states for each of the original states, and the transition model is more complex: there are four possible transitions at each point in time, corresponding to all possible combinations of changing the phonetic state and changing the context value. The number of independent transition parameters needed for the expanded HMM is 6 times the number of original phones. The total number of independent transition and context parameters needed in the articulatory DBN is 3 times the number of phones. In the chain DBN, it is equal to the number of phones.

We tested the HMM shown in Figure 6 with the basic phoneme alphabet and two different kinds of initialization: one reflecting continuity in the context variable (analogous to the Chain-DBN), and one reflecting voicing (analogous to the Articulator-DBN). The results were 3.5 and 3.2% word-error respectively, with 255k parameters. These results indicate that the benefits of articulatory/acoustic context modeling with a binary context variable can also be achieved by using a more complex HMM model. We expect this not to be the case as the number of context variables increases.

Discussion

The presence of a context variable unambiguously improves our speech recognition results. With basic phoneme alphabets, the improvements range from 12% to 29%. Statistically, these results are highly significant; the difference between the baseline and the articulator network is significant at the 0.0001 level. With the context-dependent alphabet, we observed similar effects.

Having learned a model with hidden variables, it is interesting to try to ascertain exactly what those variables are modeling. We found striking patterns in the parameters associated with the context variable, and these clearly depend on the network structure used. The $C_0/\delta C_0$ observation stream is most strongly correlated with the context variable, and this association is illustrated for the articulator network in Figure 7. This graph shows that the context variable is likely to have a value of 1 when C_0 has large values, which is characteristic of vowels. The same information is shown for the correlation network in Figure 8; the pattern is obviously different, and less easily characterized. Although we initialized the context variable in the articulator network to reflect known linguistic information about the voicing of phonemes (on the assumption that this might be the most significant single bit of articulator state information), the learned model does not appear to reflect voicing directly.

For the networks with time-continuity arcs, the parameters associated with the context variable indicate that it is characterized by a high degree of continuity. (See Figure 9.) This is consistent with its interpretation as representing a slowly changing process such as articulator position.

Conclusion

In this paper we demonstrate that DBNs are a flexible tool that can be applied effectively to speech recognition, and show that the use of a factored-state representation can improve speech recognition results. We explicitly model articulatory-acoustic context with an auxiliary variable that complements the phonetic state

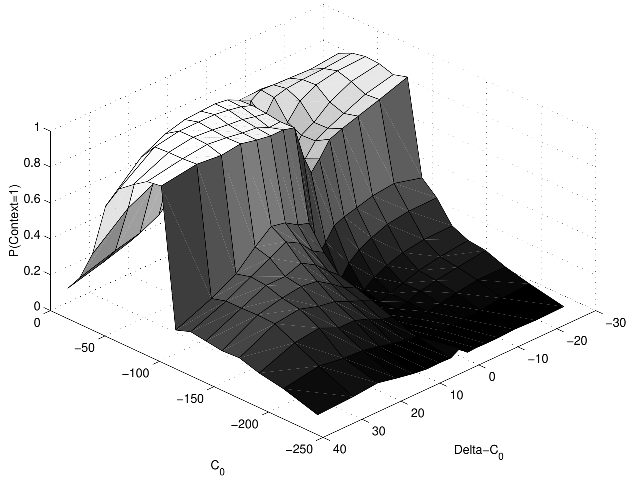

Figure 7: Association between the learned context variable and acoustic features for the articulatory network. C_0 is indicative of the overall energy in an acoustic frame. The maximum value in an utterance is subtracted, so the value is never greater than 0. Assuming that each mel-frequency filter bank contributes equally, C_0 ranges between its maximum value and about 50 decibels below maximum.

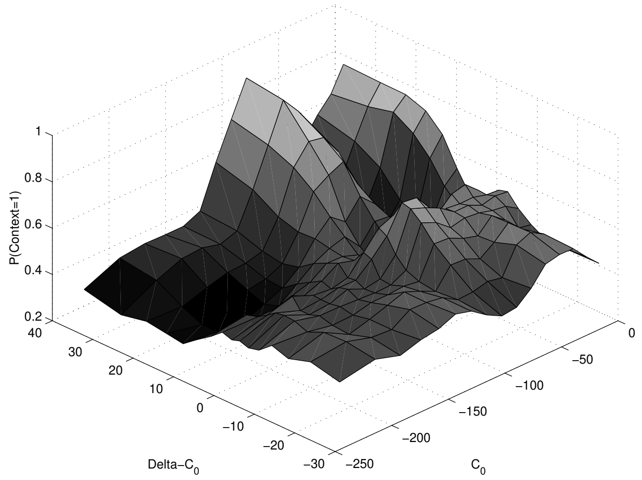

Figure 8: Association between the learned context variable and acoustic features for the correlation network. This shows a quite different pattern from that exhibited by the articulator network. (For clarity, the surface is viewed from a different angle.)

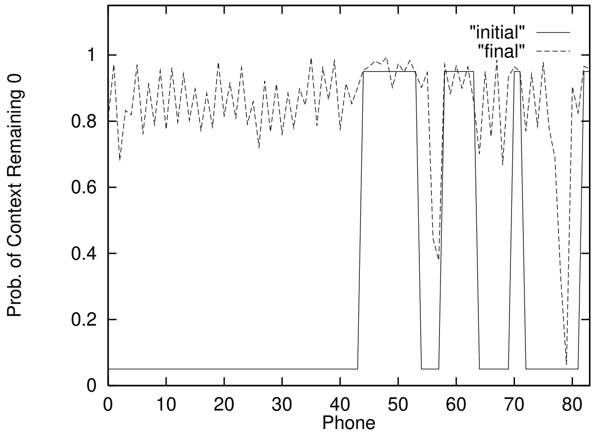

Figure 9: Learning continuity. The solid line shows the initial probability of the auxiliary state value remaining 0 across two consecutive time frames as a function of the phone. The variable was initialized to reflect voicing, so low values reflect voiced phones. The dotted line indicates the learned parameters. The learned parameters reflect continuity: the auxiliary variable is unlikely to change regardless of phone. This effect is observed for all values of the auxiliary chain. To generate our recognition results, we initialized the parameters to less extreme values, which results in fewer EM iterations and somewhat better word recognition.

variable. The use of a context variable initialized to reflect voicing results in a significant improvement in recognition. We expect further improvements from multiple context variables. This is a natural approach to modeling the coarticulatory effects that arise from the inertial and quasi-independent nature of the speech articulators.

Acknowledgments

This work benefited from many discussions with Nelson Morgan, Jeff Bilmes, Steve Greenberg, Brian Kingsbury, Katrin Kirchoff, and Dan Gildea. The work was funded by NSF grant IRI-9634215, and ARO grant DAAH04-96-1-0342. We are grateful to the *International Computer Science Institute* for making available the parallel computing facilities that made this work possible.

References

Bourlard, H., and Morgan, N. 1994. *Connectionist Speech Recognition: A Hybrid Approach.* Dordrecht, The Netherlands: Kluwer.

Davis, S., and Mermelstein, P. 1980. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28(4):357–366.

Dean, T., and Kanazawa, K. 1989. A model for reasoning about persistence and causation. *Computational Intelligence* 5(3):142–150.

Dupont, S.; Bourlard, H.; Deroo, O.; Fontaine, V.; and Boite, J.-M. 1997. Hybrid HMM/ANN systems for training independent tasks: Experiments on PhoneBook and related improvements. In *ICASSP-97*, 1767–1770. Los Alamitos, CA: IEEE Computer Society Press.

Ghahramani, Z., and Jordan, M. I. 1997. Factorial hidden Markov models. *Machine Learning* 19(2/3).

Lauritzen, S. L. 1995. The EM algorithm for graphical association models with missing data. *Computational Statistics and Data Analysis* 19:191–201.

Lee, K.-F. 1989. *Automatic speech recognition: The development of the SPHINX system.* Dordrecht, The Netherlands: Kluwer.

Peot, M., and Shachter, R. 1991. Fusion and propagation with multiple observations. *Artificial Intelligence* 48(3):299–318.

Pitrelli, J.; Fong, C.; Wong, S.; Spitz, J.; and Leung, H. 1995. Phonebook: A phonetically-rich isolated-word telephone-speech database. In *ICASSP-95*, 101–104. Los Alamitos, CA: IEEE Computer Society Press.

Rabiner, L. R., and Juang, B.-H. 1993. *Fundamentals of Speech Recognition.* Prentice-Hall.

Robinson, A., and Fallside, F. 1991. A recurrent error propagation speech recognition system. *Computer Speech and Language* 5:259–274.

Russell, S.; Binder, J.; Koller, D.; and Kanazawa, K. 1995. Local learning in probabilistic networks with hidden variables. In *IJCAI-95*, 1146–52. Montreal, Canada: Morgan Kaufmann.

Smyth, P.; Heckerman, D.; and Jordan, M. 1997. Probabilistic independence networks for hidden Markov probability models. *Neural Computation* 9(2):227–269.

Zweig, G., and Russell, S. J. 1997. Compositional modeling with dpns. Technical Report UCB/CSD-97-970, Computer Science Division, University of California at Berkeley.

Zweig, G. 1998. *Speech Recognition with Dynamic Bayesian Networks.* Ph.D. Dissertation, University of California, Berkeley, Berkeley, California.