PROBABILISTIC MODELING WITH BAYESIAN NETWORKS FOR AUTOMATIC SPEECH RECOGNITION

Geoffrey Zweig† and Stuart Russell‡

†IBM T. J. Watson Research Center, and International Computer Science Institute
‡University of California, Berkeley
gzweig@watson.ibm.com, russell@cs.berkeley.edu

ABSTRACT

This paper describes the application of Bayesian networks to automatic speech recognition (ASR). Bayesian networks enable the construction of probabilistic models in which an arbitrary set of variables can be associated with each speech frame in order to explicitly model factors such as acoustic context, speaking rate, or articulator positions. Once the basic inference machinery is in place, a wide variety of models can be expressed and tested. We have implemented a Bayesian network system for isolated word recognition, and present experimental results on the PhoneBook database. These results indicate that performance improves when the observations are conditioned on an auxiliary variable modeling acoustic/articulatory context. The use of multivalued and multiple context variables further improves recognition accuracy.

1. INTRODUCTION

Hidden Markov models are the most widely used method for doing automatic speech recognition, and are based on a very simple set of concepts and assumptions: the models are expressed in terms of phonetic states and acoustic emissions, and are parameterized by transition and emission probabilities. Specifically, an HMM expresses the probability of a segmentation of an acoustic observation stream $o_1, o_2, \ldots, o_n$ into phonetic states $q_1, q_2, \ldots, q_n$ as: $P(\mathbf{o}, \mathbf{q}) = P(q_1)P(o_1|q_1)\prod_{t=2}^{n} P(q_t|q_{t-1})P(o_t|q_t)$. Although there is a great deal of variation in the meaning assigned to the states, and in the acoustic features, there is little variation in the basic factorization of the joint probability distribution.

A Bayesian network is a more general way of expressing and computing with probability distributions [6]. With a Bayesian network, it is possible to associate an arbitrary set of variables with each speech frame, and model their joint probability distribution. Hence, it is straightforward to construct models in which phonetic state information (represented by a phone variable Q_t) is augmented with variables representing, for example, articulator positions or speech rate. Moreover, an arbitrary set of conditional independence assumptions can be used to factor the joint probability distribution. There are standard algorithms for computing with Bayesian networks, which can perform the same functions as an HMM.

It is often possible to construct an HMM that represents the same probability distribution as a Bayesian network by using states that represent the Cartesian-product of the variables in each frame (a "cross-product HMM"). However, inference with the HMM can be significantly slower [9, 3, 10]. The relationship between Bayesian networks and HMMs is discussed further in [5, 9, 3].

We have implemented a system for isolated word recognition with Bayesian networks. In previous work [11], we reported results for the PhoneBook database [8] showing relative improvements in the word error rate of between 12 and 29% (depending on network topology and initialization) with a binary auxiliary variable representing acoustic/articulatory context. This paper reviews the Bayesian network structures that are necessary for speech recognition, and presents new results showing that the use of multivalued and multiple context variables results in a further improvement. Additionally, we present results in which the network is structured for unsupervised utterance clustering.

2. BAYESIAN NETWORKS

2.1. Definition

A Bayesian network expresses a joint probability distribution over a set of random variables, and consists of:

1. A set of random variables $X_1, \ldots X_n$.
2. A directed acyclic graph in which each variable appears once. The immediate predecessors of a variable X_i are referred to as its parents, with values $Parents(X_i)$. The joint probability distribution is factored as:

$$P(X_1 = x_1, \ldots, X_n = x_n) = \prod_{i=1}^{n} P(X_i = x_i | Parents(X_i)).$$

3. A representation of the required conditional probabilities. When the variables are discrete, a tabular representation is convenient. For real-valued acoustic observations, Gaussian mixtures can be used.

Temporal processes are modeled with a variant referred to as dynamic Bayesian networks (DBNs) [1] . In a DBN, a set of variables is associated with each frame, and the complete set of variables consists of the union of all these subsets. The graph structure is repeating, and the conditional probabilities associated with analogous variables in different frames are tied.

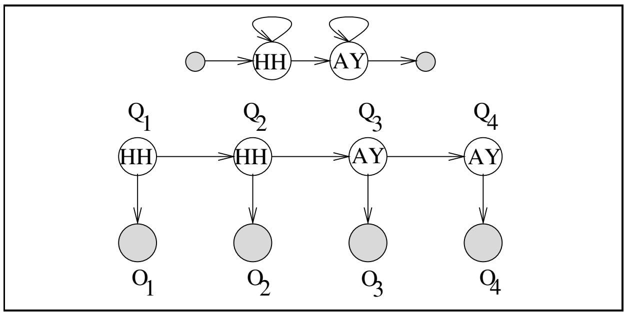

Figure 1: An HMM model of the word "hi" (top), and a conceptual DBN representation (bottom) for a four-frame utterance. Nodes represent states in the HMM, and variables in the DBN. Shaded nodes represent initial and final states in the HMM, and observed (acoustic) variables in the DBN. Arcs represent transitions in the HMM, and conditioning relationships in the DBN. The values assigned to the DBN state variables correspond to one particular path through the HMM: two time steps in /HH/, and two in /AY/. This DBN model is inadequate because it will assign nonzero probability to assignments that do not correspond to paths in the HMM, and cannot represent parameter tying (see text).

There is a distinction between variables with known values (observation variables) and variables whose values are unknown (hidden variables). We will refer to a set of assignments to the observation variables by $\mathbf{o}$, and to a set of assignments to the hidden variables by $\mathbf{q}$. As discussed in Section 2.3, there are procedures for computing $P(\mathbf{o}) = \sum_{\mathbf{q}} P(\mathbf{o}, \mathbf{q})$ (analogous to summing over all paths through an HMM), and $\mathbf{q}^* = \arg\max_{\mathbf{q}} P(\mathbf{o}, \mathbf{q})$ (analogous to Viterbi decoding). There are also EM algorithms.

2.2. Isolated Word ASR Networks

We begin the discussion of DBN word models by relating DBNs to HMMs. Figure 1 shows an HMM word model, and a schematic DBN representation. There are several things to note. First, the DBN is explicit about time: there is a separate set of variables for each frame. Secondly, the two diagrams must be read in very different ways: the HMM diagram represents a stochastic finite state automaton, whereas the DBN diagram represents conditional independence relations between variables. In the HMM, the nodes represent states and the arcs transitions; in the DBN, the nodes represent variables, and the arcs represent conditioning.

The basic idea behind the DBN representation is to create a one-to-one correspondence between assignments of values to the hidden variables, and paths through the HMM. The two representations should assign equal probabilities to analogous paths/assignments. Unfortunately, the schematic DBN of Figure 1 and will associate nonzero probability with variable assignments that do not correspond to valid paths through the HMM (for example, when all the state variables are simply assigned the value /HH/).

The DBN of Figure 1 also does not accurately represent parameter tying. To see this, consider a left-to-right word model of the word "digit": /D IH JH IH T/. The occurrence of the /IH/-/JH/ transition requires that $P(Q_t = /JH/ \mid Q_{t-1} = /IH/) \neq 0$, whereas the occurrence of the /IH/-/T/ transition requires $P(Q_t = /JH/ \mid Q_{t-1} = /IH/) = 0$. (Otherwise, the second /IH/ could be followed by another /JH/ rather than /T/. Therefore, the two occurrences of /IH/ must be treated as different states, precluding parameter tying.

Figure 2 shows a DBN that solves the various problems associated with the simpler representation. The position variables represent the state in an HMM word model at each time frame. The word model is assumed to be a simple left-to-right model so position i is always followed by $i + 1$. (In general, arbitrary finite-state word models can be represented [10].) The phone variables represent the corresponding phone labels, and the transition variables explicitly represent when there are transitions between phones.

Figure 2 shows a representative assignment of values for the word "digit." Thus position 1 maps into /D/, position 2 into /IH/, and so forth. The probability of a transition is conditioned on the phone, thus encoding a distribution over phone durations. Depending on the value of the preceding transition variable, the position variable in a frame either retains its previous value or increases by 1. The "end-of-word" variable is assigned the arbitrary value of 1, and the conditional probabilities are defined as $P(EOW = 1 | Position \neq 5 \; or \; Transition \neq 1) = 0$. This ensures that all assignments end with a transition out of the last emitting state in the word. The explicit representation of phone labels and transitions allows for parameter tying. The context variables are not required to emulate an HMM, but improve performance. With the context variable as shown, the network is similar to factorial HMMs [3].

With this representation, it is possible to assign the conditional probabilities so that there is a one-to-one correspondence between assignments of values to the DBN variables, and paths through an HMM [10]. The transition and emission probabilities are encoded in the conditional probabilities associated with the transition and observation variables. All the other conditional probabilities are either 0 or 1, and reflect deterministic relationships between the variables.

With the basic machinery required to emulate an HMM established, a variety of more interesting network structures can be tested. In particular, variables can be introduced to represent acoustic context, articulator positions, noise sources, speech rate, and other factors [10]. With a DBN, the corresponding joint probability distribution can be factored in very general ways, and the models can be tested without writing new code.

2.3. Algorithms

Bayesian networks are useful in ASR because there are algorithms for performing the same tasks that can be solved for HMMs, while at the same time working with more general models of probability distributions. The algorithms are based on dynamic programming, and involve computations similar to the forward and backward recursions of HMM inference. However, the algorithms are somewhat more involved because they work with arbitrary network structures. In the worst case, the inference algorithms have the same time-complexity as inference with a cross-product HMM, but usually the conditional independence relations inherent in a network render them more efficient. For example, in [3] a class of networks is discussed where the observations are conditioned on M independent state chains (as opposed to 1 in Figure

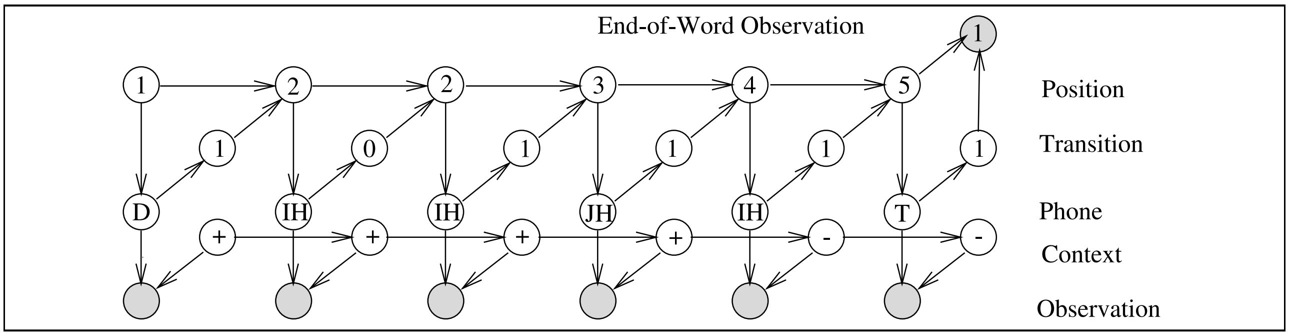

Figure 2: An improved DBN representation of an HMM. This DBN will associate a probability of 0 with hidden variable assignments that do not correspond to paths through the HMM. It also directly represents parameter tying, so for example, positions 2 and 4 will have identical behavior with respect to transition and emission probabilities because they both correspond to /IH/. The context variables are not needed to emulate an HMM, but improve performance. In this picture, they are are assigned values representative of voicing. The last /IH/ is unvoiced due to feature spreading.

1). Assuming each state variable has K values, inference with a DBN is $O(MK^{M+1})$ as opposed to $O(K^{2M})$ for a cross-product HMM representation.

The implementation used in this work is based on the algorithm of [7], with efficiency improvements described in [10]. The sufficient statistics required for EM can be gathered directly from the results of the inference procedure [4].

3. EXPERIMENTAL RESULTS

This section presents results for the PhoneBook database, which is a large collection of telephone-quality isolated-word utterances chosen to exhibit coarticulatory effects [8]. The data was processed in 25ms frames overlapped by 2/3, to generate MFCCs and their deltas. Following cepstral mean subtraction, the MFCCs and deltas were vector quantized separately into two eight-bit data streams. C_0 and delta-C_0 were each quantized to four bits and concatenated to form a third eight-bit data stream.

Training, tuning and test sets are as in [2]. There were $19,421$ training utterances, $7,291$ tuning utterances, and $6,598$ test utterances. There was no overlap between the training and testing vocabularies or speakers. The database has a vocabulary of about $8,000$ words, divided into subsets of about 75 words each; the test task consisted of selecting among the word models in a single subset. Our word models were based on the context-independent phone transcriptions provided with the database.

Previous work [11] established that the use of a single binary context variable (as in Figure 2) can significantly improve performance, and Table 1 indicates that the use of multivalued context variables and multiple context chains (two per frame) further improves performance. Using two binary context chains was as good or better than using a single 4-valued context chain. The factored representation is preferable because it has only 2 independent context-transition parameters as opposed to 15.

It is not surprising that the ability to model context improved recognition performance. However, the use of a context variable differs significantly from the use of context-dependent phones: context-dependent phones encode a-priori knowledge about expected acoustics, based on the surrounding phone labels, and are insensitive to the acoustics of individual utterances. A context variable as in Figure 2 captures information about the surrounding acoustics as observed on an utterance-by-utterance basis. For example, consider simple left-to-right word models with context-dependent phones. The sequence of phones will be the same for all utterances of a particular word. In contrast, a context variable can switch unpredictably between values.

Table 2 shows results using a context-dependent phonetic alphabet based on diphones (see [11]). Doubling the number of parameters by using a context-dependent alphabet produced a greater improvement than using a context variable with context-independent phones. However, the use of both kinds of context did the best (2.6% word error rate). The combination was better than a system with about the same number of parameters that simply used twice as many context-dependent phones.

Our results improve on the 4.1% result reported in [2] for a hybrid ANN-HMM system with continuous-valued feature vectors (rather than VQ) and using the same word transcriptions as in our work. Howej(to)j(exhibit)g(coarticulatory)f(ef)o(fects)h([8].)26 b(The)13 b(data)g(w

context variable was most strongly related to C_0 and delta-C_0; the relationship is illustrated in Figure 3 for a single binary context variable and 4-state phone models. The context variable tends to have a value of 0 when delta-C_0 is near 0, or slightly negative. The pattern is the same for 3-state phone models, and with the context-dependent alphabet, but different for other network topologies.

To illustrate the ease with which Bayesian networks can be used to perform different tasks, we configured the network of Figure 2 to do unsupervised utterance clustering. This is done by constraining the auxiliary variables to "copy" the previous value, which can be done with appropriate conditional probabilities. A clustering network produced a word-error-rate of 4.6%, and the resulting

States per Phone	Number of Context Variables	Context Variable Arity	Total System Params	Word Error Rate
3	0 (HMM)	-	96k	5.4%
3	1	2	191k	4.1%
3	1	3	287k	4.0%
3	1	4	383k	3.8%
3	2	2	383k	3.6%
4	0 (HMM)	-	127k	4.8%
4	1	2	254k	3.6%
4	1	3	381k	3.5%
4	1	4	508k	3.2%
4	2	2	508k	3.2%

Table 1: Results for networks with one and two context variables per frame; $\sigma \approx 0.25\%$.

Network	Parameters	Error Rate
CDA-HMM	257k	3.2%
CDA-Chain-BN	515k	2.6%
CDA-HMM	510k	3.1%

Table 2: Test results with a context-dependent alphabet (CDA). The first two CDA results used 336 phones; the last CDA result used less frequently occurring phones and had a size of 666. The CDA-Chain-BN has the topology of Figure 2. $\sigma \approx 0.20\%$.

clusters show interesting patterns with respect to both speaker and word characteristics. In a Viterbi decoding, 75% of the female utterances were placed in cluster 0, and 82% of the male utterances were placed in cluster 1. Moreover, words beginning with a liquid consonant (e.g. laundromat and livelihood) tended to be assigned to cluster 0, while words ending in a liquid consonant (e.g. pathological and unethical) were associated with cluster 1. For both these word characteristics, about 69% of the utterances were placed in the predominant cluster.

4. CONCLUSION

Bayesian networks are a well-principled and flexible way of representing and reasoning with probability distributions. This paper applies Bayesian networks to isolated word ASR, and presents experimental results that show that the use of an auxiliary context variable can improve recognition performance. We are currently extending the methodology to continuous speech recognition and more complicated network structures.

5. ACKNOWLEDGMENTS

This work benefited from discussions with Jeff Bilmes, Brian Kingsbury, Su-Lin Wu, Nikki Mirghafori, Eric Fosler-Lussier, and Nir Friedman. It 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.

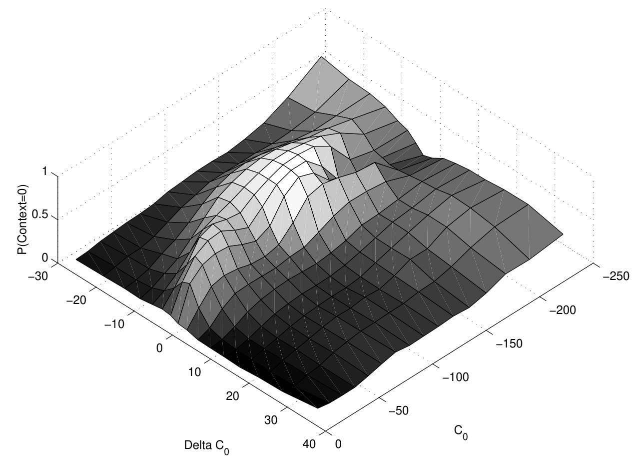

Figure 3: Association between the learned context variable and acoustic features for the network of Figure 2. Assuming that each mel-frequency filter bank contributes equally, C_0 ranges between its maximum value and about 50 decibels below maximum.

6. REFERENCES

1. Thomas Dean and Keiji Kanazawa. Probabilistic temporal reasoning. In *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI-88)*, pages 524–528, St. Paul, Minnesota, 1988. American Association for Artificial Intelligence.
2. S. Dupont, H. Bourlard, O. Deroo, V. Fontaine, and J.-M. Boite. Hybrid HMM/ANN systems for training independent tasks: Experiments on PhoneBook and related improvements. In *ICASSP-97*, pages 1767–1770. IEEE Computer Society Press, 1997.
3. Z. Ghahramani and M. I. Jordan. Factorial hidden Markov models. *Machine Learning*, 29(2/3), 1997.
4. D. Heckerman. A tutorial on learning with Bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research, Redmond, Washington, 1995. Revised June 1996.
5. H. Lucke. Which stochastic models allow Baum-Welch training? *IEEE Trans. on Signal Processing*, 44(11):2746–2755, 1996.
6. Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, California, 1988.
7. Mark Peot and Ross Shachter. Fusion and propagation with multiple observations. *Artificial Intelligence*, 48(3):299–318, 1991.
8. J. Pitrelli, C. Fong, S. Wong, J. Spitz, and H. Leung. Phonebook: A phonetically-rich isolated-word telephone-speech database. In *ICASSP-95*, pages 101–104. IEEE Computer Society Press, 1995.
9. P. Smyth, D. Heckerman, and M. Jordan. Probabilistic independence networks for hidden Markov probability models. Technical Report MSR-TR-96-03, Microsoft Research, Redmond, Washington, 1996.
10. Geoffrey Zweig. *Speech Recognition with Dynamic Bayesian Networks*. PhD thesis, University of California, Berkeley, Berkeley, California, 1998.
11. Geoffrey Zweig and Stuart Russell. Speech recognition with dynamic Bayesian networks. In *AAAI-98*, 1998.