AUTOMATIC SPEECH RECOGNITION USING DYNAMIC BAYESIAN NETWORKS WITH BOTH ACOUSTIC AND ARTICULATORY VARIABLES

Todd A. Stephenson[1,2] *Hervé Bourlard*[1,2] *Samy Bengio*[1] *Andrew C. Morris*[1]

[1]Dalle Molle Institute for Perceptual Artificial Intelligence (IDIAP), Martigny, Switzerland
[2]Swiss Federal Institute of Technology at Lausanne (EPFL), Lausanne, Switzerland

ABSTRACT

Current technology for automatic speech recognition (ASR) uses hidden Markov models (HMMs) that recognize spoken speech using the acoustic signal. However, no use is made of the causes of the acoustic signal: the articulators. We present here a dynamic Bayesian network (DBN) model that utilizes an additional variable for representing the state of the articulators. A particular strength of the system is that, while it uses measured articulatory data during its training, it does not need to know these values during recognition. As Bayesian networks are not used often in the speech community, we give an introduction to them. After describing how they can be used in ASR, we present a system to do isolated word recognition using articulatory information. Recognition results are given, showing that a system with both acoustics and inferred articulatory positions performs better than a system with only acoustics.

1. INTRODUCTION

In state-of-the-art automatic speech recognition (ASR), hidden Markov models (HMMs) utilize two random variables x_t and q_t, the acoustics and the hidden state, respectively. The likelihood of the acoustic sequence given the model is then calculated from the emission probabilities and the transition probabilities, respectively:

$$P(x_t|q_t) \tag{1}$$

$$P(q_t|q_{t-1}). \tag{2}$$

The inclusion of a third random variable, a_t, to represent the articulatory information was shown to be beneficial in speaker-dependent ASR (Zlokarnik, 1995). That work replaced the emission probability in (1) with

$$P(x_t, a_t|q_t). \tag{3}$$

In one of his tests, the actual articulator values were replaced with estimated values that a multi-layer perceptron (MLP) provided based on the acoustics. This system performed better than an acoustics only system.

The present paper investigates the use of dynamic Bayesian networks (DBNs) for incorporating articulatory data with acoustic data in ASR, building upon the groundwork done in Zweig and Russell (1998); Zweig (1998). So far, DBNs have not been used extensively in speech recognition. The final chapter of Zweig (1998) outlines as a future research area the incorporation of articulatory information into DBNs. Our current work is taking this path. DBNs are well suited for handling articulatory information because

1. they model the *causal relationships* among the variables, and
2. they can readily handle *missing data*.

First, in standard ASR, the one causal relationship that is modeled is that of phonetic state→acoustics, as given in (1); that is, the part of the phone that the speaker wants to say causes certain acoustics. A DBN can expand this relationship to also model the more realistic causal relationship of articulators→acoustics:

$$P(x_t|a_t, q_t); \tag{4}$$

it can also model the dependency of the articulator on the phonetic state and on the previous articulator:

$$P(a_t|a_{t-1}, q_t). \tag{5}$$

Second, while this articulatory data will be available during the training of the DBN, it will not realistically be available in a production setting. Nevertheless, DBNs can readily handle missing data so that during recognition the DBN is able to infer the distribution of the missing articulatory positions, given the observed acoustics.

2. BAYESIAN NETWORKS

A Bayesian network (Pearl, 1988) (see Figure 1) is composed of the following three items:

- the **variables** $\boldsymbol{X}$ that are being modeled,
- a **directed acyclic graph** (DAG) where there is a one-to-one mapping between the vertices in the graph and the variables,
- a conditional, prior **probability distribution** for each variable X_i, as given in (6) below.

Each variable has a probability distribution conditioned on the variables who have edges pointing to it (i.e., its parents), as illustrated in Figure 1. In other words, a variable's probability distribution is

$$P(X_i|\text{parents}(X_i)). \tag{6}$$

Note that the edges themselves do not carry any probability distributions. The joint probability of all the variables $\boldsymbol{X}$ is then assumed to be the product of all the (local) probability distributions within the variables:

$$P(\boldsymbol{X}) = \prod_{\forall X_i} P(X_i|\text{parents}(X_i)). \tag{7}$$

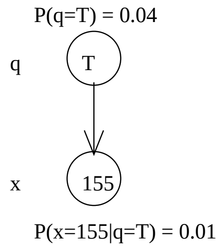

Figure 1: A Bayesian network (representing one time-frame) for ASR. Possible probabilities for the given values are provided. This is for the simplified case of phoneme recognition.

A set of observations $\boldsymbol{O}$ may be assigned to a subset of the variables in the Bayesian network. The variables that are left unobserved have an uncertainty associated with them as to what their values are. Each does have its prior probability distribution, as given in (6), but needs to have its *posterior* probability distribution inferred:

$$P(X_i|\text{parents}(X_i), \boldsymbol{O}) \tag{8}$$

The junction tree algorithm (Peot and Shachter, 1991) is an algorithm that can be used to infer these posterior probability distributions. A version of it, tailored to the needs of ASR, can be found in detail in Zweig (1998). The junction tree algorithm is similar to the Baum-Welch algorithm used in HMMs (Rabiner and Juang, 1993) in that it works with variables λ and π, which are analogous to the α and β variables, respectively, used in HMMs:

$$\begin{aligned} \lambda_j^i &= P(\boldsymbol{O_i^-}, \boldsymbol{O_i^0}|X_i = j) && (9)\\ \pi_j^i &= P(\boldsymbol{O_i^+}, X_i = j), && (10) \end{aligned}$$

where $\boldsymbol{O_i^-}$ are the observations below X_i in the junction tree, $\boldsymbol{O_i^0}$ is any observation for X_i itself, and $\boldsymbol{O_i^+}$ are all the remaining observations.

Equations (9) and (10) can then be used to compute the likelihood of the model as well as the marginal posterior probability for each variable, given the observations:

$$\forall i, P(\boldsymbol{O}) = \sum_j \lambda_j^i \pi_j^i \tag{11}$$

$$\forall i, P(X_i|\boldsymbol{O}) = \frac{\lambda_j^i \pi_j^i}{\sum_j \lambda_j^i \pi_j^i} \tag{12}$$

Dynamic Bayesian networks (Dean and Kanazawa, 1988) (DBNs, see Figure 2) are an extension of Bayesian networks for modeling dynamic processes, in this case a process over time. A regular Bayesian network is replicated for each time slice. Edges are then added between desired variables in neighboring time slices. When the node for a variable takes in a connection from a previous time frame, it then has to expand the number of variables in its conditional probability distribution by one to accommodate the possible values for the variable from the previous time frame.

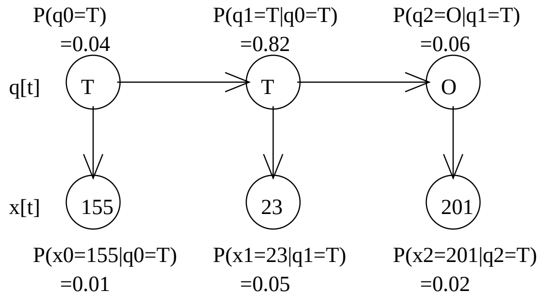

Figure 2: A dynamic Bayesian network (representing three time-frames), using the Bayesian network in Figure 1 as its base. For each successive time frame, a possible value for the phonetic state variable q_t is given as well as a possible value for the acoustic emission, x_t. Possible probabilities for the given values are also provided. This also is for the simplified case of phoneme recognition.

3. ISOLATED WORD RECOGNITION WITH DBNS

3.1. Acoustics-based recognition

Bayesian networks are most easily used in problems where each variable is discrete. For ASR, this means that the speech signal needs to be quantized. This also means that instead of Gaussian distributions being used for calculating the emission probabilities, discrete probability tables are used. The DBN in Figure 2 is for doing simple phoneme recognition. A DBN for doing isolated word recognition is illustrated in Figure 3 (further extensions for DBNs, which are not currently addressed in this work, such as language modeling, noise modeling, speaking rate modeling, etc. can be found in Zweig (1998)); it uses the following deterministic and stochastic variables for acoustics-based recognition (for explanation of the **Articulator** variable, see Section 3.2):

- Deterministic
  - **Position** refers to the current position in the word model. It takes values $1, \ldots, N$, where N is the maximum length of a word model.
  - **Phone** refers to which phone is associated with the current **Position**.
- Stochastic
  - **Transition** refers to whether a transition is being taken out of this phone. It has only two possible values: true or false.
  - **Acoustics** refers to the speech signal. In the case of multiple acoustic streams, it can be replicated for each stream for each time frame. The number of values it takes is the size of the codebook for the stream.

The increased complexity of Figure 3 over Figure 2 enforces which phones come in which order in the model (which is the responsibility of the **Position** and **Phone** deterministic variables).

The topology of the DBN differs significantly from that of HMMs. In HMMs, the transition probabilities are encoded by the edges between the state variables; in

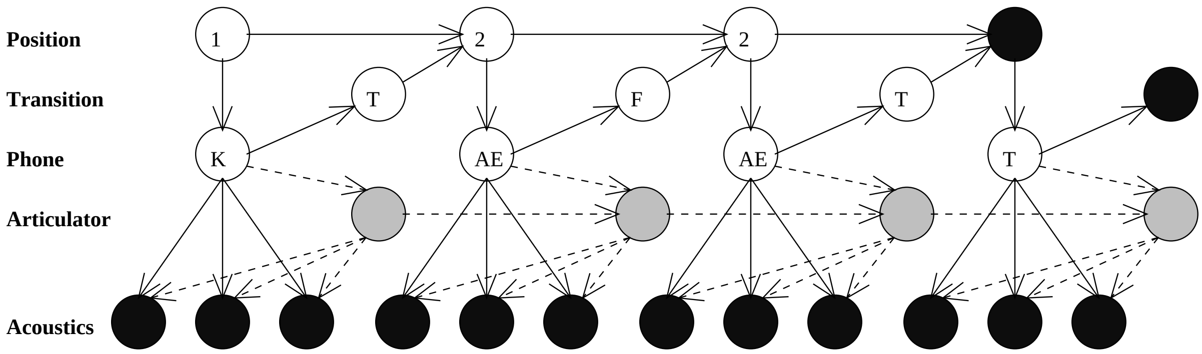

Figure 3: Based on Zweig (1998), a dynamic Bayesian network for isolated word recognition, covering four time steps. This DBN models the word "cat", pronounced using three phonemes: /k/-/æ/-/t/. This is for the acoustics/articulatory-based recognition; acoustics-based recognition uses the same model but with the **Articulator** variable and its dashed edges removed. The black vertices (the **Acoustics** and the final **Position** and **Transition** variables) are always observed. The grey vertices (the articulators) are observed in training (when available) but not in normal recognition.

DBNs the transition probabilities are encoded within the **Transition** variable. Furthermore, the legal sequence of phones in an HMM is encoded by concatenating the different phone models. In DBNs, the legal sequence of phones is determined by using the **Position** and **Phone** variables; that is, the **Phone** variable will deterministically indicate which phone can occur at each position in the word model (e.g., for Figure 3, $P(\text{Phone} = \text{K}|\text{Position} = 1) = 1$, $P(\text{Phone} = \text{Æ}|\text{Position} = 2) = 1$, $P(\text{Phone} = \text{T}|\text{Position} = 3) = 1$, with all other values of $P(\text{Phone}|\text{Position})$ equal to 0).

3.2. Acoustics/Articulatory-based recognition

In Figure 3, the **Acoustics** variable models the emission probability using (1) (if the **Articulator** variable and its dashed edges are ignored). This is interpreted as saying that the **Phone** that is being pronounced causes certain acoustics. A more plausible model would incorporate the direct causes of the acoustics, i.e., the articulators, giving (4). The dependency of a_t on both a_{t-1} and q_t is given by (5). This is represented graphically by the DBN in Figure 3 when the articulator variable and dashed connections are utilized.

The exact same algorithms are used for training of and recognition with the acoustics/articulatory DBN as when using an acoustics only DBN. This is because the DBN algorithms are independent of the topology of the DBN. These algorithms can also be used if part of the data is missing. This is vitally important during recognition. While it is reasonable to have observed articulatory information available during the training phase, articulatory observations generally will not be available during recognition. The DBN can readily handle missing data because during recognition it is able to infer the distribution of the missing articulatory positions, given the observed acoustics.

4. EXPERIMENTS

Using the University of Wisconsin X-ray Microbeam Speech Production database (Westbury *et al.*, 1994), we did experiments on speaker-independent, task-dependent, isolated word recognition. The speech is recorded at 21739 Hz with a recording of selected articulator positions (lower lip, upper lip, four tongue positions, lower front tooth, and lower back tooth) at approximately 146 Hz (6.866 ms between samples). Of the 48 speakers in the database, eight were randomly selected to be in the test set; of the remaining 40 speakers, eight were randomly selected to be in the validation set with the remaining 32 speakers comprising the training set. All three lists were constructed as to be gender-balanced. There are different tasks that the speakers were asked to do. For this work, we chose to use the "Citation Words" tasks, where the speaker reads a list of single words, separated by pauses. Using a segmentation produced by a forced alignment at IDIAP with an HTK system (Young *et al.*, 1999), the set of words for each Citation Words task were cut into individual files with some surrounding silence. The lexicon size was 106 words; some of the words were repeated multiple times by the same speaker, giving an average, across all of the data, of about 260 utterances per speaker. Thirty-nine monophones were used in addition to beginning and ending silence. Three states were used for each monophone and silence being modeled.

Twelve mel-frequency cepstral coefficients (MFCCs) plus C_0, the energy coefficient, were extracted per window from the speech, using a Hamming window of 20.598 ms with successive windows shifted by 6.866 ms. This shift rate was chosen so as to have one articulatory observation per window. There were 26 filterbanks with a preemphasis coefficient of 0.97. Energy normalization as well as cepstral mean subtraction were performed. The delta (i.e., first derivative) coefficients for all 13 MFCC coefficients were used as well.

The cepstral coefficients are then quantized using K-means clustering. Four codebooks are generated from the training data: a 256 value codebook for the 12 MFCC coefficients, a 256 value codebook for the 12 MFCC delta coefficients, a 16 value codebook for the C_0 coefficient, and a 16 value codebook for the C_0 delta coefficient. The C_0 and the C_0 delta values are concatenated bitwise in the DBN to give a single 256 value variable.

Likewise, the articulatory values are also quantized, using K-means clustering. The measurements of the eight articulators are used for the codebook. Occasionally (22% of the frames, across all of the data), an articulator value was not recorded for some time slices; in these cases, the

	WER	# Param.
Acoustics Only (baseline)	9.8%	31488
2 Discrete Articulatory Values	8.5%	62976
4 Discrete Articulatory Values	7.7%	126690
8 Discrete Articulatory Values	8.4%	257070

Table 1: Recognition results, given as Word Error Rate (WER), for models trained on the training set, with recognition performed on the validation set. The number of free parameters is given in the final column.

	WER
Acoustics Only (baseline)	8.6%
4 Discrete Articulatory Values	7.8%

Table 2: Recognition results, given as Word Error Rate (WER), for models trained on both the training set and the validation test with recognition performed on the test set. Only the best acoustics/articulatory system from Table 1 was used.

whole vector was thrown out and not used in any part of the experiments. One codebook was generated to represent all eight articulator positions. Various values for the size of the codebook are presented in this paper: two, four, and eight. The baseline DBN system did not use an articulatory variable; with such a configuration, it was theoretically equivalent to a standard ASR HMM.

We used an in-house DBN program for training and testing the models. This program has previously been tested against the performance given by an standard discrete HMM implemented using HTK, and the recognition performance between the two, acoustics-only systems were comparable on a large reference database (Phonebook). All models were trained using expectation-maximization (EM) training; after the log likelihood increased by less then 1% from the previous iteration, one more maximization step was done before termination. Dirichlet priors of 0.1 were used on all probabilities to prevent any from becoming 0. Except where noted, recognition was then performed using only the acoustics from the validation set (the articulators were ignored and thus treated as hidden). Results are given in Table 1 for a system trained on the training set with recognition on the validation set. As can be seen, the word error rate is improved when articulatory information is added.

Using the optimal number of discrete articulatory values on the validation set given in Table 1, we then started the experiments over using only the baseline system (acoustics only) and the best acoustics/articulatory system (with four articulatory values). However, this time all codebook generation and DBN training were done on the combination of the training set and the validation set. Recognition was then done on the test set. The results are given in Table 2. The results of these recognition tests are the true estimates of the two systems' performances on new data as the test set was not used previously to select any parameters for either system.

	WER
4 Discrete Articulatory Values	7.6%

Table 3: Using *observed* articulator values, recognition results, given as Word Error Rate (WER), for models trained on both the training set and the validation test with recognition performed on the test set. Only the best acoustics/articulatory system from Table 1 was used.

5. CONCLUSIONS

We presented a system for doing isolated word recognition that infers articulatory information from the acoustics and uses this information for enhanced recognition. Articulatory observations are provided during training, but articulatory information is only inferred from the acoustics as a probability distribution during recognition. The results of Table 2 show that the performance of an acoustics/articulatory system was superior to that of the traditional, acoustics only system, 7.8% verses 8.6%, respectively. Thus, a 10% reduction in the word error rate was achieved. For comparison, Table 3 gives recognition results with the articulatory variable observed (even though this is not a realistic scenario). The small difference of the recognition performance of the articulatory/acoustic system with missing articulatory data verses observed articulatory data, 7.8% verses 7.6%, respectively, suggests that the DBN is able to fairly accurately infer the articulatory positions from the acoustics.

ACKNOWLEDGEMENTS

This work is supported by the Swiss National Science Foundation under grant # 21-53960.98. We also thank Sacha Krstulovic for sharing his work previously done on the database.

References

Dean, T. and Kanazawa, K. (1988). Probabilistic temporal reasoning. In *Proceedings of the Seventh National Conference on Artificial Intelligence*.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc., San Francisco, California, revised second printing edition.

Peot, M. A. and Shachter, R. D. (1991). Fusion and propagation with multiple observations in belief networks. *Artificial Intelligence*, **48**, 299–318.

Rabiner, L. and Juang, B.-H. (1993). *Fundamentals of Speech Recognition*. PTR Prentice-Hall, Inc., Englewood Cliffs, NJ.

Westbury, J. R., Turner, G., and Dembowski, J. (1994). *X-ray Microbeam Speech Production Database User's Handbook*. Waisman Center on Mental Retardation & Human Development, University of Wisconsin, Madison, WI, first edition.

Young, S., Kershaw, D., Odell, J., Ollason, D., Valtchev, V., and Woodland, P. (1999). *The htk book*. Entropic, Ltd., Cambridge, UK, htk version 2.2 edition.

Zlokarnik, I. (1995). Adding articulatory features to acoustic features for automatic speech recognition. *The Journal of the Acoustical Society of America*, **97**(5), 3246. Abstract 1aSC38.

Zweig, G. and Russell, S. (1998). Probabilistic modeling with Bayesian networks for automatic speech recognition. In R. H. Mannell and J. Robert-Ribes, editors, *ICSLP '98 Proceedings*, volume 7, pages 3011–3014, Sydney.

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