An Introduction to Dynamic Bayesian Networks for Automatic Speech Recognition

Marcus Uneson
Lund University

Abstract

Bayesian Networks are a particular type of Graphical Models, providing a general and flexible framework to model, factor, and compute joint probability distributions among random variables in a compact and efficient way.

For speech recognition, a BN permits each speech frame to be associated with an arbitrary set of random variables. They can be used to augment well-known statistical paradigms such as Hidden Markov Models by decomposing each state into several variables, outside acoustics representing for instance articulators or speech rate. Factoring joint probability distributions may potentially lead to more meaningful state representations as well as more efficient processing. Bayesian networks have also been applied for language modeling.

Bayesian networks are rather new in the field of automatic speech recognition. Within the scope of a term paper, we provide an introduction to their main properties and give some examples of their current use.

1. Introduction

1.1 Purpose and Scope

The purpose of this paper is to provide a first introduction to Bayesian Networks (BNs), with some examples of their use in Automatic Speech Recognition (ASR).

'Introduction' is indeed a keyword – on one hand, it should be possible to follow the paper without any particular familiarity with the subject; on the other, anyone who wishes just a little bit more than a high-level view will have to dig in the references, or the references of the references.

Given the very limited scope, the presentation of the theoretical foundations is mostly absent. For the same reason, there are no discussions at all of algorithms or implementational details. While many of these subjects – learning algorithms, for one – certainly are central and interesting (and, in some cases, huge), only some pointers can be given here.

1.2 Organization

The paper is organized as follows. In Section 2, the problem of statistical speech recognition is very briefly summarized, and a typical current system based on HMMs (Hidden Markov Models) is outlined. Some recurring shortcomings of such a system are mentioned.

Section 3 presents Bayesian Networks (BNs) in general. First, it introduces Graphical Models (GMs), which represent stochastic processes as graphs in a flexible and unifying way (for instance, well-known statistical machinery such as HMMs may also be readily represented as GMs). Then it turns to BNs, the particular subtype of GMs which is the main topic of this paper, their representation, and their usage. Finally, it outlines Dynamic BNs (DBNs), an extension of BNs to handle time series (such as speech signals) and sequences.

Section 4 tries to tie (D)BNs and ASR together. First, some advantages of DBNs to the current method of choice, HMM, are commented, and a way of expressing HMMs as DBNs is shown. Finally, some examples of how BNs have been used in recent approaches to DBN-based speech recognition are given: for representing articulators, pitch/energy, and language models.

1.3 DBN-ASR Researchers

Bayesian networks have only very recently been applied to ASR. While the field is exciting and rather active (Zweig's seminal PhD thesis, Zweig (1998), is one of the most cited papers on ASR in the latest years), most things remain to be done, and it is too early to identify any best practices. Much of the groundwork has been done at UCLA, Berkeley. Researchers with particular interest and important publications in the field include among others Geoffrey Zweig, Jeff Bilmes, Khalid Daouid, Murat Deviren, Karen Livescu, Todd Stephenson.

2. Background

2.1 Statistical Speech Recognition

The overview given in this section is based on Holmes & Holmes (2001), Zweig (1998), Zweig & Russell (1998), Bilmes (1999), Deviren (2004). However, on this general level, comparable presentations could be found in most any work on ASR.

2.1.1 Overview

Human speech perception can be thought of as the mapping of an acoustic time signal to meaningful linguistic units. Following this view, Automatic Speech Recognition (ASR) is the task of defining an association between the acoustic signal and the linguistic units in such a way that it can be implemented on a computer.

The acoustic signal holds strong random components, and perhaps it is only logical that most recent approaches employ a probabilistic framework. The question to be answered by such a system is "what is the most probable linguistic representation W, given an acoustic waveform A?", or, more mathematically put, to find $W^* = \arg\max_w P(W|A)$. By Bayes' rule and by considering P(A) as a constant, this formula may be rewritten as

$$W^* = \arg\max_w P(W \mid A) = W^* = \arg\max_w P(A \mid W)P(W)$$

In doing so, the problem is split into two independent subproblems, which may be attacked separately: one acoustic (estimating the a posteriori probability P(A|W) of a particular sequence of acoustic observations A, given a sequence of linguistic units W) and one linguistic (estimating the a priori probability of W, P(W). This approach is known as 'source-channel model'.

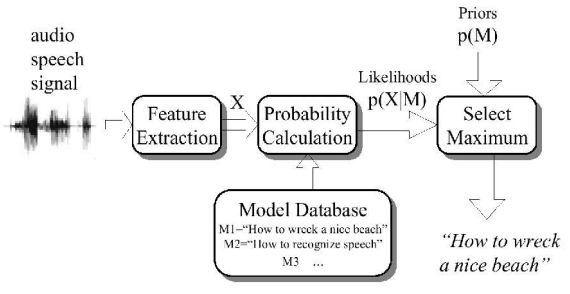


Figure 1. A generic statistical ASR system (from Bilmes (1999)).

2.1.2 A typical statistical ASR system

Figure 1 illustrates the general method used by most statistical speech recognition systems. The speech signal A is sampled, and the waveform representation is converted to a set of feature vectors, one for each time slice (typically windows of 20-40 ms width, sampled each 10-20 ms). The features (generically named X in the figure and treated as a random variable) are chosen to compactly capture as much as possible of the linguistically relevant features present in the waveform, while at the same time discarding as much as possible of the irrelevant ones (dialect, speech rate, voice quality, age and sex of the speaker, etc). Long-established feature vectors include linear predictive coding coefficients (LPCCs); more recently, the human auditory system has influenced more perceptually relevant representations such as mel-frequency cepstrum coefficients (MFCCs).

In the next step, the feature vectors are evaluated against each acoustic utterance model M in a model database, pre-built from an annotated speech corpus during a training phase. The evaluation against acoustic models yields a number of likelihoods p(X|M), which are combined with the prior linguistic probabilities p (M) of each utterance. For some applications, the most probable combination of these two is taken as the target (i.e., the intended linguistic representation W from Section 2.1.1); for others, a number of high-scoring candidates may be kept to be later evaluated on a discourse level (not shown in Figure 1).

The suitable design of the acoustic model database depends on the application. For small vocabularies, the models may correspond to words or even entire utterances. For larger, real-world vocabularies (perhaps several tens of thousands of words), the models will necessarily have to be composed by submodels, for instance on syllable or allophone level. The latter representation is flexible and theoretically compact – any utterance in a language can typically be represented symbolically with 40-60 allophones – but transitions between allophones involve many difficult approximity influences. A typical strategy is to include the transition phases into the submodels (as in diphone or triphone models). This partly solves the coarticulation problem, but only at the cost of increased perplexity.

The evaluation against the acoustic models does not usually happen directly. Instead, most methods use one or more hidden (i.e., non-observable) variables to represent the current state of the speech generation process (the pronunciation model), and maintains a probability distribution for each observation in a given state (the acoustic model).

2.2 HMM-based systems

2.2.1 Properties of HMM-based systems

In an ASR system based on Hidden Markov Models (HMMs), the predominant technique for the last decades, two variables are used for each time slice: one visible O_t (for the observation at time t) and one hidden Q_t (for the system's state at time t). The latter is usually identified with the current phonetic unit. In addition to the (stationary) observation probabilities associated with each of the N states, an HMM keeps a representation of the initial state probability $P(s_1)$ and the state transition probabilities, $P(s_{t+1}|s_t)$ for each state (also stationary; as in all Markovian models, the state is assumed to be independent of the previous history).

The joint probability P(q, o) of a certain state sequence $q = q_1q_2q_3...q_T$ given a certain observation sequence $o = o_1o_2o_3...o_T$ may be written

$$P(o,q) = P(q_1)P(o_1 \mid q_1)\prod_{t=2}^{T} P(q_t \mid q_{t-1})P(o_t \mid q_t)$$

There are efficient ($O(n^2)$) algorithms for training the parameters (the observation and transition probabilities and the initial state probabilities); for computing the probability of a certain observation sequence; and for finding the most probable state sequence given an observation sequence (Rabiner 1989).

2.2.2 Weaknesses of HMM-based systems

There are many existing HMM-based ASR systems, some of them commercial, which exhibit good recognition performance in conditions closely resembling those of the training settings. However, in more varying conditions, they are generally not very robust, and performance may decrease drastically when facing common real-world (real-word?) phenomena, such as noisy environments, differing speech rates, dialectal variation, and non-native accents (Holmes & Holmes (2001), Zweig (1998)).

While many state-of-the-art systems with some success employ different adaption techniques to overcome such difficulties, there are arguably certain inherent restrictions to the entire approach. Most importantly, HMMs encode all state information in just one single variable – to a HMM-based system, speech is little more than a finite sequence of atomic elements, each taken from a (likewise finite) set of phonetic states. This view excludes potentially important generalizations about the peculiarities of the speaker – temporary ones, such as position and movement of the articulators (lips, tongue, jaw, voicelessness, nasalization, etc), as well as more permanent, such as the speaker's sex and dialect (Zweig (1998)).

3. Bayesian Networks

3.1 Graphical models

3.1.1 What is a graphical model?

In a graphical model (Whittaker 1990; sometimes called "probabilistic graphical model"), a stochastic process is described as a graph. The graph contains a qualitative part, its topography, and a quantitative part, a set of conditional probability functions. In fact, the entire model can be thought of as "a compact and convenient way of representing a joint probability distribution over a finite set of variables" (after Bengtsson (1999)).

The nodes in a graphical model represent a set of (hidden or observed) random variables $X = \{X_1...X_n\}$, whose ranges may be continuous or discrete. The edges encode the central concept of *conditional independence* between variables.

3.1.2 Conditional independence

If A, B, C are random variables, A is said to be *conditionally independent of C given B* (written $A \perp C \mid B$) iff $P(A|B, C) = P(A|B)$; intuitively, this means that if we know B, the evidence of C does not influence our belief in A. Exactly how conditional independence is asserted in the edges of a graphical model may vary. See Section 3.2 for an example from Bayesian networks, and Stephenson (2000) or Murphy (2001) for useful overviews.

Conditional independence assertions are extremely important. First and foremost, they allow local inferences. This means that calculations

of joint probability distributions of conditionally independent subsets of variables can be performed separately, reducing complexity. Furthermore, such conditionally independent subsets can be combined to form complex structures in a modular way.

Disciplined use of conditional independence assertions whenever depencies aren't strictly necessary (with the definition of 'necessary' to be decided by the task at hand) will result in sparse networks, i.e., networks with relatively few edges per node. This will in itself create at least three important advantages compared to fully-connected models (Bilmes (2000)):

- sparse network structures have fewer computational and memory requirements;
- sparse networks are less susceptible to noise in training data (i.e., lower variance) and less prone to overfitting (the smaller the freedom – here, number of random variables – the less risk that meaningless regularity in the data will be treated as significant); and
- the resulting structure might reveal high-level knowledge about the underlying problem domain that was previously drowned out by many extra dependencies.

3.1.3 Properties and subtypes of GMs

GMs are very versatile. Combining useful traits from graph theory and probability theory, they offer an intuitive, visual representation of conditional independence, efficient algorithms for fast inference, and strong representational power. In Michael Jordan's words, they "provide a natural tool for dealing with two problems that occur throughout applied mathematics and engineering – uncertainty and complexity" (Jordan (2004)).

Many important current models, such as HMMs, can be expressed as particular instances of GMs; and a central algorithm such as the Baum-Welch algorithm for HMM training is just a special case of GM inference (Bilmes (2000)). Indeed, the GM framework is flexible enough to subsume many existing important techniques and by its proponents it is greeted as a unifying statistical framework, greatly facilitating experimentation with new statistical methods (not only for ASR). Toolkits for GMs which permits such experimentation are increasingly available; for ASR, see Bilmes & Zweig (2002).

There are many different types of GMs, each with a different formal semantic interpretation and a concomitant different idea of the way conditional independence is encoded in the graph topology. Broadly, GMs can be divided into subclasses according to the graphs they are built upon: the most important are undirected GMs, where edges denote correlation (Markov random fields); and directed acyclic GMs, where edges informally denote causality (Bayesian networks). The former type is popular among physicists, while the second is much used in AI and, increasingly, ASR research (Murphy (2001)). The rest of this paper will only deal with Bayesian networks.

3.2 Bayesian networks

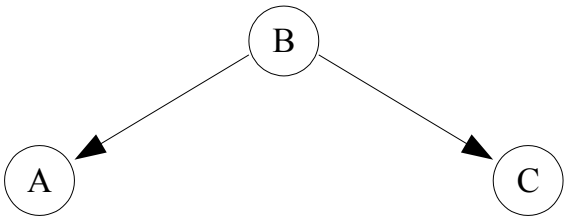
3.2.1 What is a Bayesian network?

A Bayesian network is a particular kind of GM: a graph where the each node denote one random variable $X_i \in \{X_1, X_2, ..., X_n\}$ and the (absence of) edges imply conditional independencies. The hallmark of a BN is that it is built on a directed and acyclic graph. To each variable, a prior conditional probability distribution is associated. Most of the theory for BNs is due to Pearl (1988).

Figure 2 gives an example of a simple Bayesian network, consisting of three binary (T[rue]/F[alse]) variables with associated probabilities.

In this BN, the random variable A is conditionally dependent on B, but not on C, i.e. $P(A|B, C) = P(A|B)$. Another way of formulating the same fact is to say that the value of C is irrelevant for the local probability P(A, B). The table specifies the conditional probability distribution of each node X_i for each combination of values of its immediate predecessors, which more commonly are known as the node's *parents.* In this paper, the parents of a given node are denoted $Pa(X_i)$, following Stephenson (2000).

Note that the BN in Figure 2 does not express that A and C are totally independent, but only that B encodes any information from A that influences C and vice versa. For instance, let A denote "Dan wears his coat" and C "Dan has ice in his hair". While A and C are likely to be correlated, they might still be conditionally independent given for instance B, "it is a cold day".



B	P(B=T) = 0.6	P(B=F) = 0.4
A	P(A=T|B=F) = 0.7	P(A=F|B=T) = 0.8
	P(A=T|B=T) = 0.2	P(A=F|B=F) = 0.3
C	P(C=T|B=T) = 0.5	P(C=F|B=T) = 0.5
	P(C=T|B=F) = 0.1	P(C=F|B=F) = 0.9

Figure 2. A three-variable Bayesian network (from Stephenson (2000), with example conditional probability table added).

3.2.2 Graph topology and causality

The graph topology accounts for the qualitative part of the BN, i.e., which variables are conditioned on which. The quantitative part consists of some kind of numerical or functional representation of the conditional probabilities involved. For discrete ranges, the probability distribution is typically stored in a *node probability table;* as a collection, these make up a *conditional probability table* (such as the one in Figure 2). For continuous variables, Gaussian mixtures may be used (Bengtsson (1999)).

The directed edges of a BN provide an informal representation of causality – an edge normally goes from a cause to a consequence. This notion is useful for constructing BNs by hand or for interpreting automatically derived BNs. However, the reservation of informality is important. While it is true that a given graph only corresponds to a certain joint probability distribution, the converse is not true: a given joint probability distribution may be described with many different graphs (i.e., the JPD may be *factorized* in many different ways). For instance, by Bayes' rule $P(A|B) = P(B|A)P(A)/P(B)$ some edges could be reversed and hence have inverted causal interpretations (Stephenson (2000)).

3.2.3 Joint probability calculations

The total JPD of a BN can by the chain rule of probabilities be expressed as

$$P(x_1, x_2, x_3, \ldots, x_n) = P(x_1|x_2, x_3, \ldots, x_n)P(x_2|x_3, \ldots x_n)\ldots P(x_{n-1}|x_n)P(x_n)$$

However, this form does not consider the possible simplifications due to assumed conditional independencies. If we do, we may from each factor exclude all conditionally independent variables, i.e, only consider the parents $Pa(X_i)$ of each node. The JPD may then be simplified to

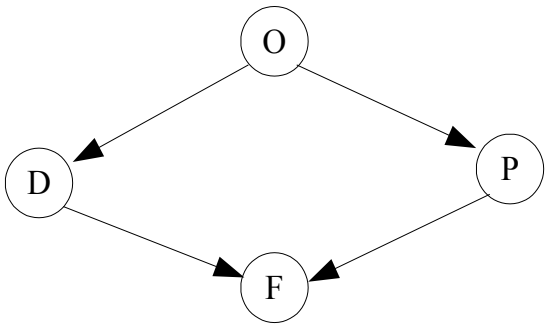
$$P(x_1, x_2, \ldots, x_n) = \prod_1^n P(Xi \mid Pa(Xi)).$$

For instance, for the simple network in Figure 2, by the chain rule the JPD P(A, B, C) can be calculated as P(A|B, C)P(B|C)P(C). However, considering the conditional independence $P(A \perp C \mid B)$, the JPD may more efficiently be written (and calculated) as P(A, B, C) = P(B)P(A|B)P(C|B).

Factorizing the JPD in this way, the number of multiplications will be of order $O(n2^t)$, where $t \ll n$ is the maximum number of incoming edges of any node. For sparse networks, this is much more efficient than the chain rule, which requires $O(2^n)$ (Murphy (2001)).

3.2.4 A BN example

Consider the BN in Figure 3, only slightly more complex. It consists of four binary (true/false) random variables and their associated conditional probability table (example modified from Murphy (2001)):



O: Ann oversleeps
D: Ann drives too fast
P: Ann parks illegally
F: Ann is fined

O	P(O=T) = 0.1	P(O=F) = 0.9
D	P(D=T|O=T) = 0.2	P(D=F|O=T) = 0.8
	P(D=T|O=F) = 0.05	P(D=F|O=F) = 0.95
P	P(P=T|O=T) = 0.4	P(P=F|O=T) = 0.6
	P(P=T|O=F) = 0.15	P(P=F|O=F) = 0.85
F	P(F=T|D=T, P=T) = 0.1	P(F=F|D=T, P=T) = 0.9
	P(F=T|D=T, P=F) = 0.03	P(F=F|D=T, P=F) = 0.97
	P(F=T|D=F, P=T) = 0.08	P(F=F|D=F, P=T) = 0.92
	P(F=T|D=F, P=F) = 0.001	P(F=F|D=F, P=F) = 0.999

Figure 3. A Bayesian network with four variables

A BN may incorporate some of our prior beliefs about the presence or absence of causal relations. In this case, for instance, F may be caused by D or

P (or both); O may cause D and/or P. (O probably also have causes, but they fall outside the universe of this particular network.) The values of the prior beliefs may be trained from date, if available, or estimated by experts.

3.2.5 Using a Bayesian network: Inference

A BN may be used for many things. For instance, in Figure 3, we already have (prior) conditional probabilities for D and P given O. From those, we can (without observing anything) calculate the unconditional, or *marginal*, probability of D and P as well (in which case O is said to be 'marginalized out').

More often, however, a BN is used for *probabilistic inference*, the computation of the probabilities of a set of random variables after having gained information (*evidence*) about the values of some other set of variables. Before we receive evidence, the BN represents our a priori belief about the system that it models; after we have done so, the network may be updated to denote our a posteriori beliefs.

Inferences are possible in the 'causal' (top-down reasoning, from root to leaf or cause to effect) as well as in the 'diagnostic' direction (bottom-up reasoning, from leaf to root or effect to cause) (Murphy (2001)).

In the situation pictured in Figure 3, we may for instance look out of the window and note that Ann indeed parks illegally today. That is, we receive *hard evidence* that P = true (or 'the variable P is instantiated to True'). The evidence may be used to calculate (updated) probabilities for causes (O) as well as effects (F) of P. In other cases, we may only receive *soft* evidence, that is, the added knowledge that the probability distribution of P has been changed (for instance, to accommodate new hard evidence for some other variable). See Fenton (2004) for some other tutorial examples of BN usage.

The updating of probabilities in light of new evidence, be it soft or hard, is known as *belief propagation*. This is a demanding task (NP-hard, in the general case); in fact, belief propagation as a concept has been around much longer than efficient algorithms to perform it. Among the recent breakthroughs of the field are new algorithms built on groundwork by Lauritzen and Spiegelhalter. These permit quick updates of many BNs occurring in practice, containing many thousands of nodes. Zweig and Russel (1998) report to have trained models with up to 500000 parameters. See Murphy (2002) for an overview of the algorithms.

3.2.6 Learning BNs

A BN is completely defined by its topology and its conditional independence parameters. It is possible to learn both from data, although learning topology (graph structure) is much harder than numeric parameter optimization (i.e., what for well-known models such as HMMs is referred to as 'training'). Furthermore, the structure to be learned may contain hidden variables, which additionally adds to the task.

Learning BNs is a huge subject, and it falls entirely outside the limited scope of this paper. See for instance the overviews Murphy (2002), Heckerman (1995), and Heckermans chapter in Jordan (1998).

3.2.7 Tools, other applications

Although Bayesian networks are a rather recent approach in ASR, they have been put to use in many other different areas. Other practical applications include spam filtering for e-mail (Sahami (1998)), troubleshooters for non-experienced computer users, and medical diagnosis systems (Stephenson (2000)).

3.3 Dynamic Bayesian Networks

The Bayesian networks discussed so far all specify a certain point in time – they are static. They need to be extended in order to account for temporal processes such as speech (or, more generally, sequences of any kind, for instance DNA or n-grams – for the latter, see Section 4.3). This is commonly accomplished by a straightforward extension.

Dynamic Bayesian Networks (DBNs) are Bayesian networks which include directed edges pointing in the direction of time (Bilmes (2001)). The structure and parameters are assumed to repeat for each time slice (i.e., the process is assumed to be stationary), so the conditional probabilities associated with $X_i[t]$, $1 <= t <= T$, are tied. In fact, DBNs can be seen as "unrolling" a one-frame network for T time steps (Friedman (1998)) and adding time-dependencies, in effect creating a BN of size N x T.

If the processes modelled are assumed to be Markovian (”the future is conditionally independent of the past given the present”, i.e.,

dependency edges are only permitted between time frame t and t+1), it is enough to specify the initial network (Figure 4a) and the edges connecting two consecutive time slices (a so called 2TBN, Figure 4b) and then repeat them as necessary (Figure 4c, for four time slices).

However, DBNs can be used also to model non-Markovian processes, by permitting longer dependency edges (Deviren (2004); see Section 4.3.3 for an example).

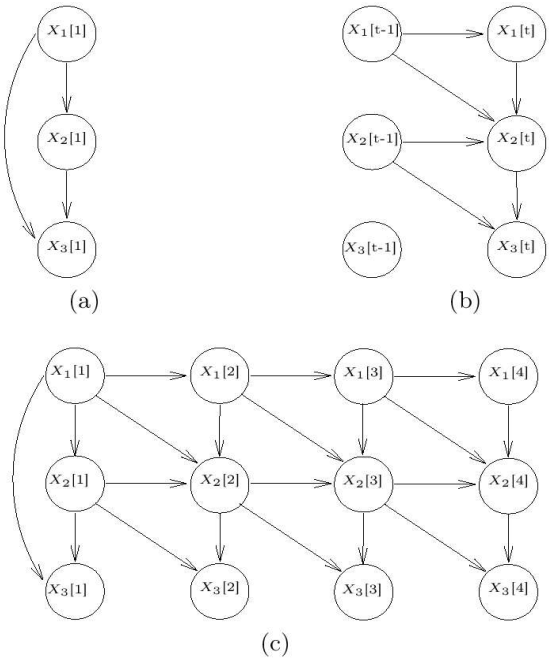


Fig. 4. Illustration of a DBN representation and the unrolling mechanism (from Deviren (2004), after Friedman (1998)) (a) Initial network. (b) Transition network. (c) Unrolled DBN for four time slices.

4. Examples of DBNs in ASR

4.1 Dynamic BNs and ASR

Most current ASR systems are based on HMMs. In comparison to these, DBNs offer a more general and flexible framework to model, factor, and compute joint probability distributions (JPDs) among random variables. In contrast to HMMs, a DBN permits each speech frame to be associated with an arbitrary set of random variables. Thus, by introducing variables representing, say, the positions of the articulators, the speaker's sex, the speech rate etc, a DBN has the potential of decomposing state representations into more meaningful components. This way of factoring JPDs may make more meaningful state representations possible, including causal relationships (and, although this claim has been made before, as a side effect perhaps make better use of the expert phonetic knowledge, once again inviting phoneticians into a field long dominated by statisticians). Transitional behaviours can be described with submodels sharing variables over time, which is a useful representation of coarticulation. Furthermore, DBNs may have exponentially fewer parameters than standard HMMs. This permits parameter estimation with less data and higher computational efficiency, and missing data is handled gracefully (Zweig (1998); Daoudi (2002); Deviren (2002, 2004)).

4.2 Emulating an HMM-based ASR

According to its proponents, DBNs offer many advantages to HMMs. Still, it may be useful to start out with a DBN-based emulation of a plain HMM-based system. In this way, we may expect the same performance as that of an ordinary HMM system (at the cost of somewhat higher complexity and perhaps, but not necessarily, some computational overhead). Once this machinery is in place, the richer semantic representation provided by the framework may be explored in different directions.

There are efficient and general-purpose algorithms available for inference. These algorithms do not depend of the topology of the network, and so testing different structures could largely be done by restructuring the network rather than rewriting code.

Expressing an HMM-based ASR system in a DBN can be done in several ways. A naïve approach may come out like the one in Figure 5 (bottom). Note the very different meanings of the similar symbols for the DBN and the HMM (top): variables vs. states, explicit vs. implicit time representation, conditional dependencies vs. state transitions.

This version is simple but insufficient. Non-zero probabilities will be assigned to illegal sequences (such as repeating the phone of the first state in all speech frames). Furthermore, parameters are shared between all frames. This means that reoccurring phones in a word, such as the first and second vowel in 'digit' in Figure 5, will share not only output probabilities, which is fine, but also transition probabilities, which is problematic. Instead, the first occurence should have non-zero probability for the transition to the /j/ state and

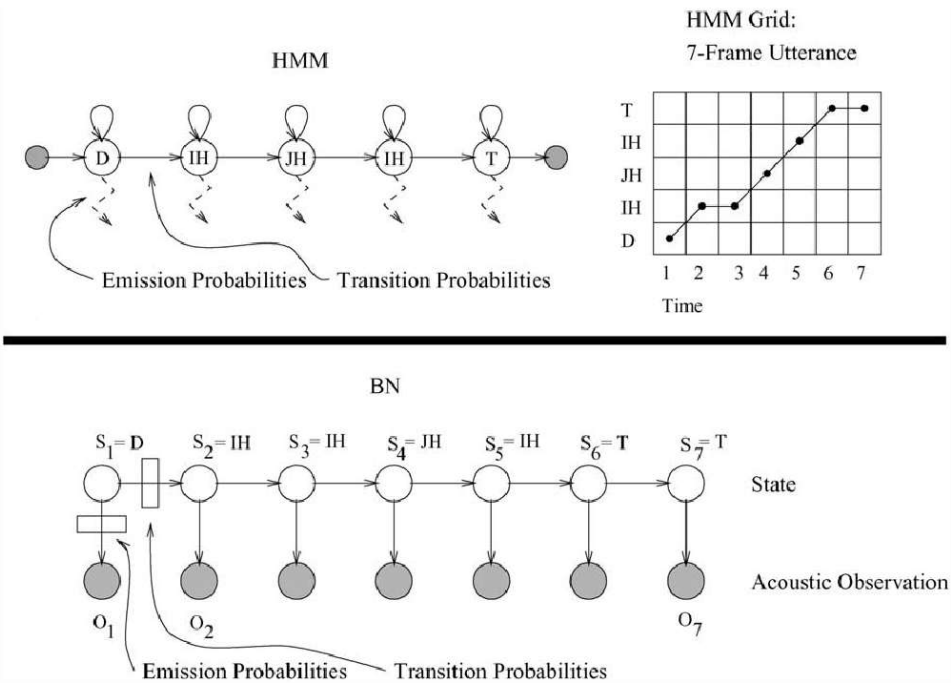


Fig. 5. An HMM model of the word 'digit' (top) and a naïve DBN representation for a particular token of the same word (from Zweig (2003)). Note the difference in semantics for the legends: In the HMM, there is only implicit time representation; unfilled nodes denote intermediary states and shaded nodes initial and final states; arcs represent state transition probabilities; dashed lines denote emission probabilities. In the DBN, the time representation is explicit (seven frames, for this token); unfilled nodes represent hidden and filled nodes observed variables; arcs denote conditioning relationships

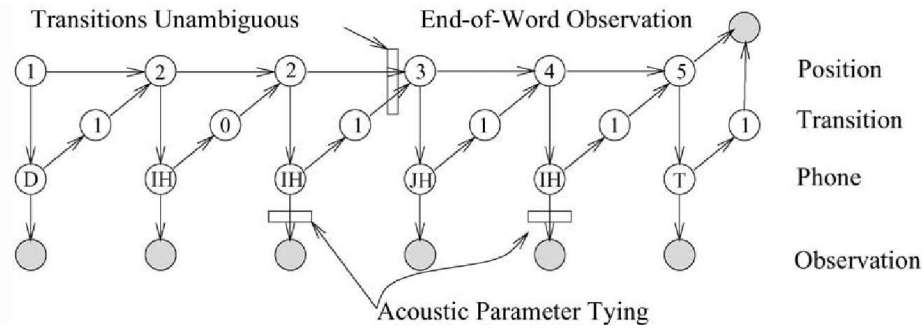


Fig. 6. An improved DBN emulation of the HMM in Figure 5 (from Zweig (2003)), here representing a six-frame occurrence of the word 'digit' . Again, filled nodes are observed, unfilled nodes are hidden

.

zero probability for the transition to the /t/ state, and the second occurence should have the figures reversed.

Zweig (1998) solves both these problems by using one (deterministic) 'Position' variable, holding the position of the phone in the word, and another (also deterministic) 'Transition' variable, holding a true/false value for the transition from one phone to another. A typical DBN implementation of a HMM may come out as something like Figure 6 (see Zweig (2003) or Zweig & Russell (1998) for details).

4.3 DBN-based models for ASR

This section mentions very briefly a few recent papers on DBN-based models for ASR. Several other important topics, such as speaker rate, speaker accent, or noise modeling, are not touched here; the examples given are only meant to convey an idea of the flexibility of the framework. Many suggestions originate from Zweig (1998). We particularly refer to this paper for more examples (and illustrations).

4.3.1 Modeling articulation with DBNs

There have been many attempts to incorporate articulatory models in HMM-based systems. Zweig (1998) mentions several reasons why this is better done with DBNs (in short: mappings and models are stochastic rather than rule-based and can be learned rather than hand-coded; and general-purpose algorithms are available for learning and inference).

The articulator variables and their ranges in fact constitute something like a stochastic phonological production theory. While such a theory certainly is very far from complete, ASR performance may be improved with a less complete version.

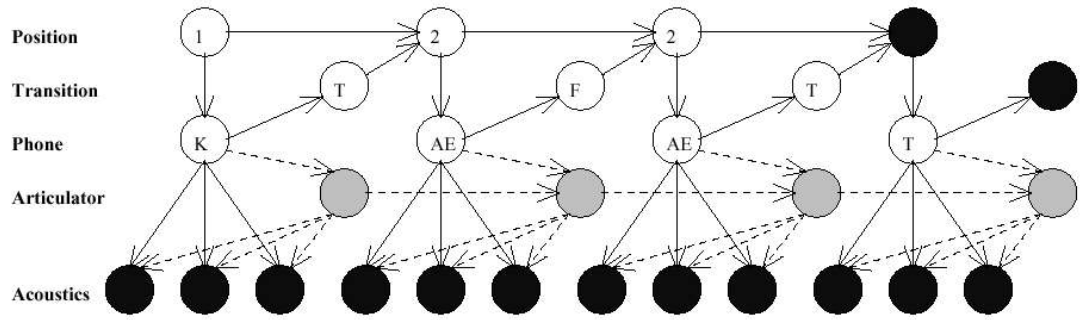


Fig. 7. A DBN for isolated word recognition (from Stephenson (2000), after Zweig (1998)). Filled nodes are always observed, grey nodes are observed in training, unfilled nodes are hidden. The DBN represents the word 'cat'. It uses an articulatory auxiliary variable and hence represents an acoustic-articulatory model. Removing the articulator variable and the dashed edges turns the DBN into a representation of an HMM acoustic model, cf Figure 6.

Stephenson et al. (2000) describes a system for isolated word recognition, employing a DBN model with one additional auxiliary variable representing the state of the articulators (see Figure 7). Measured articulatory data may be present during training but most likely are absent during recognition; however, the system does not need to observe the variable during recognition to exhibit performance improvements. The paper reports a decrease in WER by 10% (for four discrete values of the added variable), compared to the acoustics only system.

4.3.2 Modeling pitch and energy with DBNs

Stephenson also (2002) describes a (partly) successful attempt to include pitch and energy in an ASR system. While these features are notoriously difficult to model (most attempts to do so in the past have lead to performance degradation), Stephenson suggests an approach that treats them as auxiliary variables and marginalizes them out when not contributing to recognition performance (due to noise in estimation or modeling). This is rather straightforward to formulate within the DBN framework (but very much less so with HMMs). The results reported lie around the baseline, but compared to earlier attempts to consider pitch and energy, this might still be regarded as encouraging. The authors claim that the results "demonstrate the validity of the approach".

4.3.3 Language modeling with DBNs

Leaving articulation and acoustics aside, DBNs have been employed for statistical language modeling. In fact, the commonly employed n-gram and n-class models are just particular (generally non-Markovian) instances of DBNs. N-class models are particularly useful to assist n-grams in handling missing data. Deviren et al. (2004) describes an interesting attempt to combine both into a single model. For a French newspaper corpus, they report a perplexity reduction of 2.4 % when the traditional bigram model is replaced by a DBN model where a word also depends on the syntactic classes of the two previous words. All linguistic units are handled in one procedure.

The DBN structure described is built manually; however, according to the authors, a future objective is to automatically extract the best dependence relations between a word and its context, i.e., automatically inferring the model that best explains the data.

5. Conclusions

Bayesian networks and their temporal counterpart Dynamic Bayesian networks offer a rich probabilistic framework for many fields. For ASR, where they have only recently been deployed, they may among other things provide a powerful, unifying probabilistic framework; efficient, general-purpose algorithms for inference and learning; learning from data and/or expert knowledge; and representational and experimental flexibility.

References

Bengtsson, H. (1999) *Bayesian Networks - a self-contained introduction with implementation remarks*, Unpublished Master's Thesis, Lund Institute of Technology, Lund, Sweden.

Bilmes, J. (1999) *Natural Statistical Models for Automatic Speech Recognition*, Unpublished PhD thesis, International Computer Science Institute, Berkeley.

Bilmes, J. (2000) Dynamic Bayesian Multinets, *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, Stanford, July 2000

Bilmes, J. (2001) *Graphical Models and Automatic Speech Recognition* (Technical Report UWEETR-2001-0005): University of Washington, Department of Electrical Engineering.

Bilmes, J., and Zweig, G. (2002) The Graphical Models Toolkit: An Open Source Software System For Speech and Time-Series Processing, *Proceedings of the International Conference on Acoustics Speech and Signal Processing (ICASSP 2002)*, Orlando

Daoudi, K. (2002) Automatic Speech Recognition: The New Millennium, *Proceedings of the 15th International Conference on Industrial and Engineering, Applications of Artificial Intelligence and Expert Systems: Developments in Applied Artificial Intelligence*, 253-263.

Deviren, M. (2002) Dynamic Bayesian Networks for Automatic Speech Recognition, *Proceedings of the The Eighteenth National Conference on Artificial Intelligence*, Edmonton

Deviren, M. (2004) *Systèmes de reconnaissance de la parole revisités : Réseaux Bayésiens dynamiques et nouveaux paradigmes*, Unpublished PhD thesis, Université Henri Poincaré, Nancy.

Deviren, M., Daoudi, K., and Smaîli, K. (2004) Language Modeling using Dynamic Bayesian Networks, *Proceedings of the LREC 2004*, Lisbon, May 2004

Fenton, N. (2004) *Probability Theory and Bayesian Belief Nets*. Web course (accessed 20050114): http://www.dcs.qmw.ac.uk/ ~norman/BBNs/BBNs.htm.

Friedman, N., Murphy, K., and Russell, S. (1998) Learning the structure of dynamic probabilistic networks, *Proceedings of the UAI'98*, Madison, Wisconsin

Heckerman, D. (1995) *A tutorial on learning with Bayesian networks* (Technical Report MSR-TR-95-06), Redmond, Washington, USA: Microsoft Research.

Heckerman, D. (1998) A tutorial on learning with Bayesian networks, in M. Jordan (ed.), *Learning in Graphical Models*, MIT Press.

Jordan, M. I. (2004) Graphical Models, *Statistical Science*, 19, 140-155.

Murphy, K. (2001) An Introduction to Graphical Models. Unpublished tutorial. http://www.ai.mit.edu/~murphyk/Bayes/bayes.html

Murphy, K. (2002) *Dynamic Bayesian Networks: Representation, Inference and Learning*, Unpublished PhD thesis, UC Berkeley.

Pearl, J. (1988) *Probabilistic Reasoning in Intelligent Systems:Networks of Plausible Inference*, Morgan Kaufmann., San Mateo,California.

Rabiner, L. R. (1989) A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, *Proc of the IEEE*, 77, 2, 257-286.

Sahami, M., Dumais, S., Heckerman, D., and Horvitz, E. (1998) A Bayesian Approach to Filtering Junk E-mail, *Proceedings of the AAAI'98 Workshop on Learning for Text Categorization*, Madison, Wisconsin, July 27 1998

Stephenson, T. A. (2000) *An Introduction to Bayesian Network Theory and Usage* (Technical report IDIAP-RR 00-03): IDIAP.

Stephenson, T. A., Bourlard, H., Bengio, S., and Morris, A. C. (2000) Automatic speech recognition using dynamic Bayesian networks with both acoustic and articulatory variables, *Proceedings of the ICSLP 2000*, October 2000, 951-954.

Stephenson, T. A., Escofet, J., Magimai-Doss, M., and Bourlard, H. (2002) Dynamic Bayesian Network Based Speech Recognition with Pitch and Energy as Auxiliary Variables, *Proceedings of the IEEE International Workshop on Neural Networks for for Signal Processing (NNSP 2002)*

Whittaker, J. (1990) *Graphical Models in Applied Multivariate Statistics*, John Wiley & Sons Ltd, Chichester,UK.

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

Zweig, G. (2003) Bayesian network structures and inference techniques for automatic speech recognition, *Computer Speech and Language*, 17, 173-193.

Zweig, G., and Russell, S. (1998) Speech Recognition with Dynamic Bayesian Networks, *Proceedings of the The fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*, 173-180.