LEARNING WITH BAYESIAN NETWORKS

Yufeng Wu

Institute for Signal and Information Processing
Department of Electrical and Computer Engineering
Mississippi State University, Mississippi State, MS 39762
wu@isip.msstate.edu

ABSTRACT

A Bayesian network is a graphical tool which contains probabilistic relationships among various random variables. When used jointly with statistical techniques, the graphical model has several advantages for data analysis. First, a Bayesian network can be used to learn probabilistic relationships, and thus can be used in a learning system to gain the understanding of a problem domain and to predict the consequences of intervention. Second, it is an ideal representation for combining prior knowledge, which often comes in causal form, with data. Three, Bayesian statistical methods along with network structures can encode dependencies among all random variables. In this paper, we discuss methods for constructing Bayesian networks from prior knowledge and summarize Bayesian statistical methods for learning from data to improve probabilistic models of stochastic processes.

1. INTRODUCTION

A Bayesian network is a graphical tool for probabilistic relationships among a set of variables. The field of Bayesian networks, and graphical models in general, has grown rapidly over the last few years, with theoretical and computational developments in many areas. More recently, researchers have developed numerous methods for learning Bayesian networks from data. The techniques that have been developed are new and still under heavy construction, but they have been proven to be remarkably effective for some data analysis problem.

In this paper, we give a tutorial on Bayesian networks and its associated Bayesian probabilistic techniques for extracting and encoding prior knowledge from data. However, there are lots of representations available for data analysis, such as decision trees, artificial neural networks and rule bases, and there are many methods for data analysis, such as density estimation, classification, and regression. So what's the advantage of Bayesian networks?

First, a Bayesian network can be used to learn probabilistic relationships from a large amount of data for a large amount of random variables, and thus can be used in a learning system to gain the understanding of a problem domain and to predict the consequences of intervention.

Second, Bayesian networks allow one to learn about causal relationships. Learning about causal relationships are very important for at least two reasons. The process is useful when we are trying to gain understanding about a problem domain. In addition, knowledge of causal relationships allows us to make predictions in the presence of interventions. For example, a marketing analyst may want to know whether or not it is worthwhile to increase exposure of a particular advertisement in order to increase the sales of a product. To answer this question, the analyst can determine whether or not the advertisement is a cause for increased sales, and to what degree. The use of Bayesian networks helps to answer such questions even when no experiment about the effects of increased exposure is available.

Three, Bayesian statistical methods along with network structures can encode dependencies among all random variables. They can facilitate the combination of domain knowledge and data. Anyone who has performed a real-world analysis knows the importance of prior or domain knowledge, especially when the available data is scarce or expensive. Bayesian networks encode the strength of causal relationships with probabilities. Consequently, prior

knowledge and data can be combined with well-studied techniques from Bayesian statistics.

2. THE BAYESIAN APPROACH TO PROBABILITY AND STATISTICS

To understand Bayesian networks and associated learning methods, it is important to understand the Bayesian approach to probability and statistics. In this section, we provide an introduction to the Bayesian approach to probability and statistics.

2.1. The Bayesian Probability

The Bayesian probability of an event x is a person's degree of belief in that event. A Bayesian probability is a property of the person who assigns the probability (e.g., your degree of belief that the coin will land heads), however, from the classical viewpoint, a probability is a physical property of the world (e.g., the probability that a coin will land heads).

One important difference between a classical probability and a Bayesian probability is that, to measure the latter, we don't need to repeat trials. For example, suppose we are tossing a sugar cube onto a wet surface, every time the cube is tossed, its dimensions will change slightly. Thus, although the classical statistician has a hard time measuring the probability that the cube will land with a tail or head, the Bayesian statistician simply restricts his attention on the next toss, and assigns a probability. As another example, consider the question: What is the probability that the Chicago Bulls will win the championship in 2001? Here, the classical statistician must remain silent, whereas the Bayesian can assign a probability to this guess.

In general, the process of measuring a degree of belief is commonly referred to as a probability assessment. One problem with probability assessment that is the precision. Can one really say that his or her probability for event x is 0.601 and not 0.599? In most cases, he or she cannot say that, because in most cases, probabilities are used to make decisions, and these decisions are not very sensitive to small variations in probabilities. Well-establish practices of sensitivity analysis help one to know whether or not the additional precision is necessary [1]. Another problem with probabilities is the accuracy. For example, recent experience can lead to assessments that do not reflect a person's true beliefs [2]. Methods for improving accuracy can be found is the decision analysis literature [3].

2.2. Learning with Data

To illustrate how the Bayesian approach learns probabilities from the given data, let's consider a common coin. If we throw the coin into the air, it will come to land either on its tails or on its heads. Suppose we flip the coin $N + 1$ times, making sure that the physical properties of the coin and the conditions under which it is flipped remain unchangeable over time. From the first N observations, we want to determine the probability of heads on the $N + 1th$ flip.

In the classical analysis of this problem, we assume there is some physical probability of heads, which is unknown. We estimate the probability from the first N using criteria such as low bias and low variance. We then use this estimate as our probability for heads on the $N + 1th$ toss. In the Bayesian approach, we also assume there is some probability of landing with heads, but we interpret our uncertainty about this physical probability using Bayesian methods, and use the rules of probability to compute the probability of heads on the $N + 1th$ flip.

To examine the Bayesian analysis of this problem, we need some notation. We denote a variable by an upper-case letter (e.g., X, Y, X_i, Θ), and the state or value of a corresponding variable by that same letter in lower-case (e.g., x, y, x_i, θ). We denote a set of variables by a bold-face upper-case letter (e.g., $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{X_i}$). We use a corresponding bold-face lower-case letter (e.g., $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{x_i}$) to denote an assignment of state or value to each variable in a given set.

Returning to the coin tossing problem, we define Θ to be a variable whose values θ correspond to the possible true values of the physical probability. We sometimes refer to θ as a parameter. We express the uncertainty about Θ using the probability density function $p(\theta)$. In addition, we use X_i to denote the variable representing the outcome of the lth flip,and we use $D = \{X_1 = x_1, \ldots, X_N = x_N\}$ to denote the set of our observations. Thus, in Bayesian terms, the coin tossing problem reduces to computing $p(x_{N+1} | D)$ from $p(\theta)$.

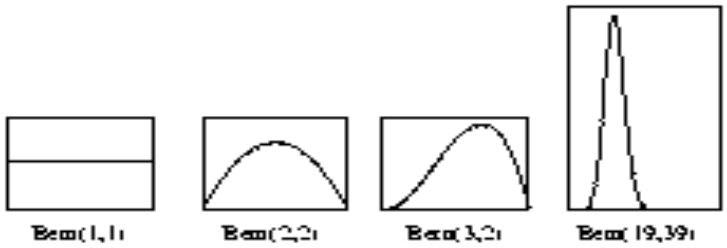

Figure 1. several commonly used beta distributions

To do so, first we use Bayes' rule to obtain the probability distribution for Θ given D and background knowledge:

$$p(\theta|D) = \frac{p(\theta)p(D|\theta)}{p(D)} \quad (1)$$

where

$$p(D) = \int p(D|\theta)p(\theta)d\theta \quad (2)$$

Next, we expand the term $p(D|\theta)$ according to the binomial distribution:

$$p(\theta|D) = \frac{p(\theta)\theta^{h}(1-\theta)^{t}}{p(D)} \quad (3)$$

where h and t are the number of heads and tails observed in D, respectively. The probability distributions $p(\theta)$ and $p(\theta|D)$ are commonly referred to as the prior and posterior for Θ, respectively. Finally we average over the possible values of Θ to determine the probability that the $N+1th$ flip of the coin will be heads:

$$\begin{aligned} & p(X_{N+1} = HEADS|D) \\ &= \int p(X_{N+1} = HEADS|\theta)p(\theta|D)d\theta \\ &= \int \theta p(\theta|D)d\theta \end{aligned} \quad (4)$$

To complete the computation, we need to assign a prior distribution for Θ. A common approach is to assume that the prior distribution is a beta distribution:

$$\begin{aligned} & (p(\theta) = Beta(\theta|\alpha_h, \alpha_t)) \\ &= \frac{\Gamma(\alpha)}{\Gamma(\alpha_h)\Gamma(\alpha_t)}\theta^{\alpha_h - 1}(1-\theta)^{\alpha_t - 1} \end{aligned} \quad (5)$$

where $\alpha_h > 0$ and $\alpha_t > 0$ are the parameters of the beta distribution, $\alpha = \alpha_h + \alpha_t$, and $\Gamma(*)$ is the Gamma function which satisfies $\Gamma(x+1) = x\Gamma(x)$ and $\Gamma(1) = 1$. The quantities α_h and α_t are often referred to as hyper-parameters to distinguish them fromthe parameter θ. The α_h and α_t must be greater than zero in order to normalize the distribution. Examples of beta functions are show in Figure 1.

The beta prior is chosen for several reasons. By Equation 3, the posterior distribution will also be a beta distribution:

$$(p(\theta|D) = Beta(\theta|\alpha_h + h, \alpha_t + t)) \quad (6)$$

We say that the set of beta distributions is a conjugate family of distributions for binomial sampling. Also, the expectation of θ with respect to this distribution has a simple form:

$$\int \theta Beta(\theta|\alpha_h, \alpha_t)d\theta = \frac{\alpha_h + h}{\alpha + N} \quad (7)$$

Hence, given a beta prior, we have a simple expression for the probability of heads in the $N+1th$ flip:

$$p(X = HEADS|D) = \frac{\alpha_h + h}{\alpha + N} \quad (8)$$

3. BAYESIAN NETWORKS

So far we are only talking about simple problems with one or a few variables. In real learning problems, we are typically interested in looking for relationships among a large number of variables, in that case, the Bayesian networks is a ideal solution. It is a graphical tool that efficiently interprets the joint probability distribution for a large set of variables. In this section, we define a Bayesian

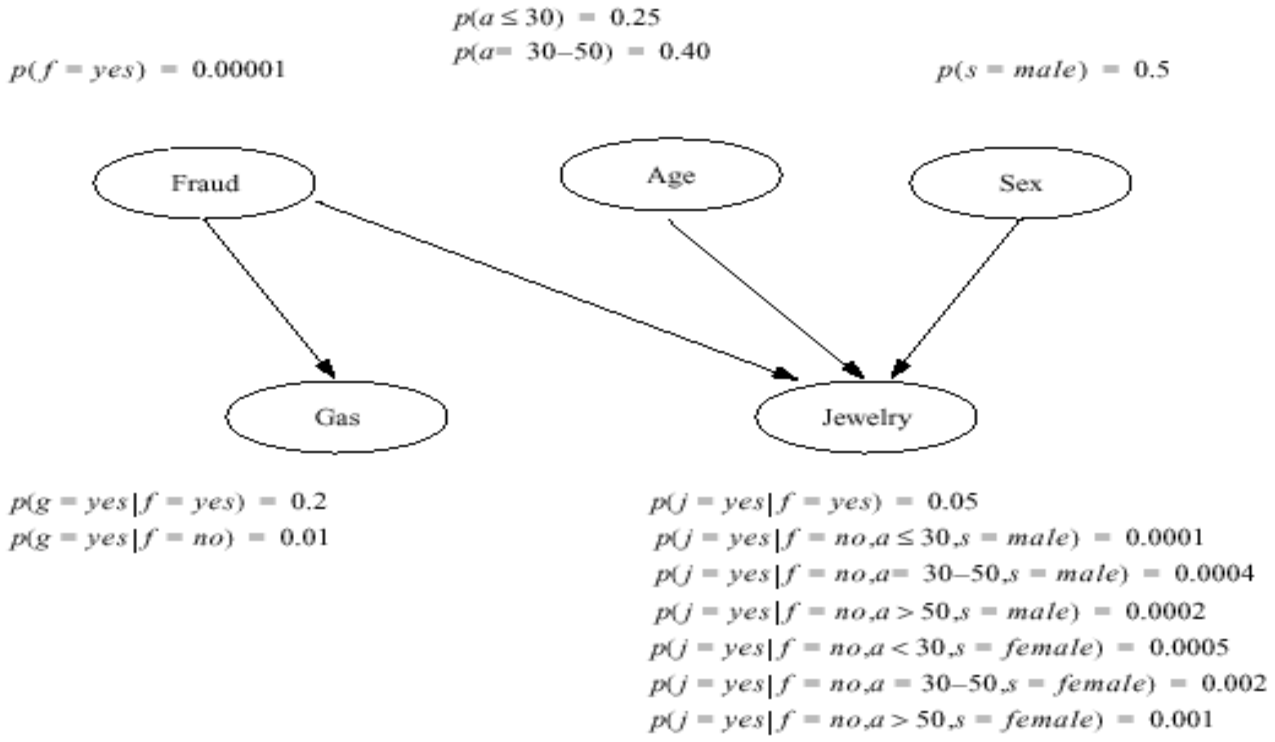

Figure 2. A Bayesian network for detecting credit-card fraud. The local probability distribution associated with a adjacent node

network and show how one can be constructed from prior knowledge.

3.1. Definition

A Bayesian network for a set of variables $\boldsymbol{X} = \{X_1, \ldots, X_n\}$ consists of a network structure S that encodes a set of conditional independence assertions about variables in $\boldsymbol{X}$, and a set of local probability distributions associated with each variable. Together, all these components define the joint probability distribution for $\boldsymbol{X}$. The network structure S is a directed aryclic graph. The nodes in the network are in one-to-one correspondence with the variables $\boldsymbol{X}$. We use X_i to denote both the variable and its corresponding node, and Pa_i to denote the parent nodes of X_i in S as well as the variables corresponding to these parents. In general, the joint probability distribution for X is given by

$$p(x) = \prod_{i=1}^{n} p(x_i | Pa_i) \tag{9}$$

The local probability distributions are the distribution corresponding to the terms in the product of Equation 9. So, the network structure S and the local distributions P decide the joint distribution $p(x)$.

3.2. Constructing a Bayesian Network

To illustrate the process of building a Bayesian network, consider the problem of detecting credit-card fraud. We begin by determining the variables to model. One possible choice of variables for our problem is Fraud (F), Gas (G), Jewelry (J), Age (A), and Sex (S), representing whether or not the current purchase is fraudulent, whether or not there was a gas purchase in the last 24 hours, whether or not there was a jewelry purchase in the last 24 hours, and the age and sex of the card holder, respectively. The states of these variables are shown Figure 2. Of course, in a realistic problem, we would include many more variables. Also we could model the states of one or more of these variables at a higher level of detail. For example, we could get Age be a continuous variable.

As part of the initial task, we must

- Correctly identify the goals of modeling
- Identify all possible observations that may be relevant to this problem
- determine what subset of these observations is worthwhile to model

- organize the observations into the variables having mutually exclusive and collectively exhaustive states.

In the next step, we will build a directed aryclic graph that shows the conditional independence among those variables. One way to do that is based on the following observations. From the chain rule of probability, we have

$$p(x) = \prod_{i=1}^{n} p(x_i | x_1, \ldots, x_{i-1}) \quad (10)$$

Now, for every X_i, there will be some subset $\prod_i \subseteq \{X_1, \ldots, X_{i-1}\}$ such that X_i and $\{X_1, \ldots, X_{i-1}\} / \prod_i$ are conditionally independent given subset $\prod_i$. That is, for any x,

$$p(x_i | x_1, \ldots, x_{i-1}) = p(x_i | \pi_i) \quad (11)$$

Thus, we get

$$p(x) = \prod_{i=1}^{n} p(x_i | \pi_i) \quad (12)$$

We see that the variables sets $(\Pi_1, \ldots \Pi_n)$ correspond to the Bayesian network parents $(Pa_1, \ldots, pa_n)$, which are able to fully describe the arcs in the network structure S. So, to determine the structure of a Bayesian network we need to order the variables somehow and determine the variables sets that satisfy Equation 12 for $i = 1, \ldots, \dot{n}$. In our example, using the ordering (F, A, S, G, J), we have the conditional independencies

$$\begin{aligned} p(a|f) &= p(a) \\ p(s|f,a) &= p(s) \\ p(g|f,a,s) &= p(g|f) \\ p(j|f,a,s,g) &= p(j|f,a,s) \end{aligned} \quad (13)$$

Thus, we obtain the network structure shown in Figure 2.

This approach has a serious drawback. If we choose the variable order carelessly, the resulting network structure may fail to show the conditional independence among the variables. For example, if we construct a Bayesian network for the fraud problem using the ordering (J, G, S, A, F), we obtain a fully ordering to find the best one. Fortunately, there is another technique to construct Bayesian networks that does not require an order among the variables. The new technique is based on two observations, first, people can often readily assert causal relationships among variables, and second, causal relationships typically correspond to assertions of conditional dependence. In particular, to construct a Bayesian network for a given set of variables, we simply draw arcs between the variable pair which has the asserted conditional dependence, from the cause to the result. For example, given the assertions that Fraud is a direct cause of Gas, and Fraud, Age and Sex are direct causes of Jewelry, we obtain the network structure in Figure 2.

In the final step of constructing a Bayesian network, we assign the local probability distribution $p(x_i | pa_i)$. In our fraud example, where all variables are discrete, we assign one distribution for X_i for each of its parent nodes. Example distributions are shown in Figure 2.

4. INFERENCE

After a Bayesian network is constructed (from prior knowledge, data, or a combination of these two), usually we want to decide the probability of a particular interest, which is corresponding to a node in our network. For example, in our credit-card fraud detecting problem, we are interested in the probability of a purchase being a fraud given all other available observations. The probability cannot be obtained directly from the network and needs to be computed. The computation of a probability of an interest is called “probabilistic inference”. In this section, we will illustrate the basic idea of computing the desired probability in a Bayesian network.

4.1. Definitions

In this section, we are going to using multinomial distribution to show how we actually proceed the probability computation in a Bayesian network.

In our case, each variable $Xi \subseteq \boldsymbol{X}$ is discrete, having r_i possible values $x_i^1, \ldots, x_i^{'i}$, and each local probability distribution is a collection of multinomial distributions, one distribution for each parent node of X_i, Namely, we assume

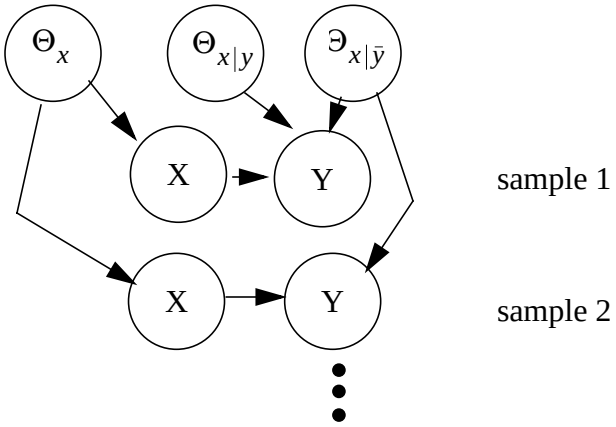

Figure 3. A Bayesian network structure describes the assumption of parameter independence for learning the parameters of the network structure $X \rightarrow Y$. Both variables X and Y are binary. We use x and $\bar{x}$ to denote the two states of X, and y and $\bar{y}$ to denote the two states of Y.

$$p(x_i^k \mid pa_i^j, \theta_i, S) = \theta_{ijk} > 0 \tag{14}$$

where $pa_i^1, \ldots, pa_i^{q_i}\left(q_i = \Pi_{X_i \subseteq pa_i} r_i\right)$ denote the configurations of pa_i, and θ_i are the parameters. For convenience, we define the vector of parameters

$$\theta_{ij} = (\theta_{ij2}, \ldots \theta_{ijr_i}) \tag{15}$$

for all i and j.

4.2. Learning Probabilities

Given this set of distributions, we can compute the posterior distribution $p\theta_s|(D, S)$ efficiently. Before we start, we need to make an assumption in order to proceed the probability computation. The assumption is that the parameter vectors θ_{ij} are independent. The corresponding mathematical interpreting is:

$$P(\theta_s \mid S) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} p(\theta_{ij} \mid S) \tag{16}$$

which was introduced by Spiegelhalter and Lauritzen (1990)

Given the joint probability distribution factors according to some network structure S, the assumption of parameter independence itself can be represented by a network. For example, the network structure in Figure 3 shows the assumption of parameter independence for $X = \{X, Y\}(X, Y \textit{ binary})$ and the hypothesis that the network structure $X \rightarrow Y$. encodes the joint probability distribution of random variable X.

Under this assumption, we can compute the posterior distribution of θ given a random sample D and network structure S

$$p(\theta_s \mid D, S) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} p(\theta_{ij} \mid D, S) \tag{17}$$

Hence, we can update each parameter vector θ independently, just as in the one variable case. Assuming each vector θ_{ij} has the prior distribution $Dir\left(\theta_{ij} \mid \alpha_{ij1}, \ldots, \alpha_{ijr_i}\right)$, we obtain the individual posterior distribution is given by the following:

$$p(\theta_{ij} \mid D, S) = Dir\left(\theta_{ij} \mid \alpha_{ij1} + N_{ij1}, \ldots, \alpha_{ijr_i} + N_{ijr_i}\right) \tag{18}$$

where N_{ijk} is the number of cases in D in which $X_i = x_i^k$ and $pa_i = pa_i^j$,

In order to compute our goal

$$p(x_{N+1} \mid D, S) \tag{19}$$

where x_{N+1} is the next case to be seen after D.

To compute this distribution, we first use the fact that the parameters remain independent given D:

$$\begin{aligned} & p(x_{N+1} \mid D, S) \\ &= \int (\prod \theta_{ijk} p(\theta_s \mid D, S)) d\theta \\ &= \prod \int \theta_{ijk} p(\theta_{ij} \mid D, S) d\theta \end{aligned} \tag{20}$$

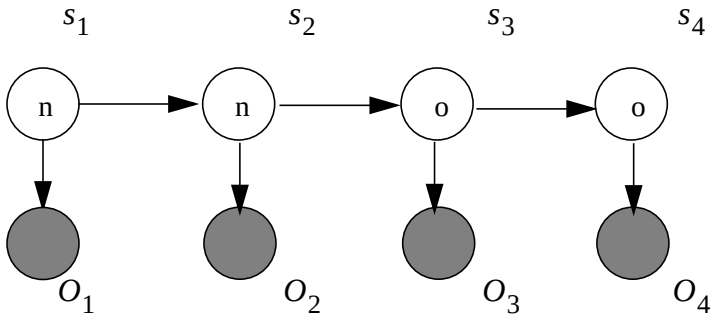

Figure 4. A Bayesian network representation of an HMM. There is a distinct state and observation variable at each point in time. A node in the graph represents a variable, and the arcs leading into a node specify the variables on which it is conditionally dependent. A valid assignment of values to the state variables for the word "no" is shown. Observation variables are shaded.

Then, we use Equation 8 to obtain

$$p(x_{N+1}|D,S) = \prod_{i+1}^{n} \frac{\alpha_{ijk} + N_{ijk}}{\alpha_{ij} + N_{ij}} \quad (21)$$

These computations are very straightforward because the multinomial distributions are in the exponential family. Computation for linear regression with Gaussian noise are equally simple [4] [5].

5. APPLICATION IN SR

In this section, we apply Bayesian networks to the speech recognition problem. The factored state representation of Bayesian networks allows us to explicitly represent acoustic context in addition to the phonetic information maintained by Hidden Markov Models (HMMs) [6] (Rabiner & Juang 1993). Furthermore, it enables us to model the multiple observation streams within single time-frames.

5.1. 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 the test data.

5.1.1. HMM

A hidden Markov Model is a simple representation of a stochastic process. 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. Furthermore, a Markovian assumption is made, so that we can obtain the following equation to compute the probability over the state sequence.

$$P(o,s) = P(s_1)P(o_1|s_1)\prod_{i=2}^{n} p(s_i|s_{i-1})p(o_i|s_i) \quad (22)$$

In the case of speech, the state variable is usually identified with the phonetic state, i.e., the current phone being pronounced.

5.1.2. 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 s, with arbitrary conditional independence assumptions. Often there is a significant decrease in the computational load if the conditional independence assumptions result in a sparse network [7] [8] [9].

More precisely, a Bayesian network represents a probability distribution over a set of random variables $\boldsymbol{V} = V_1, \ldots V_n$. 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

$$P(v_1, \ldots, v_n) = \prod_i P(v_i|Parents(V_i)) \quad (23)$$

where $Parents(V_i)$ are the parents of V_i in the graph.

5.2. Acoustic Modeling

The reason for using a Bayesian network 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 [10] [11]. The simplest approach is to augment the phonetic state variable with one or more variables that represent articulatory-acoustic context.

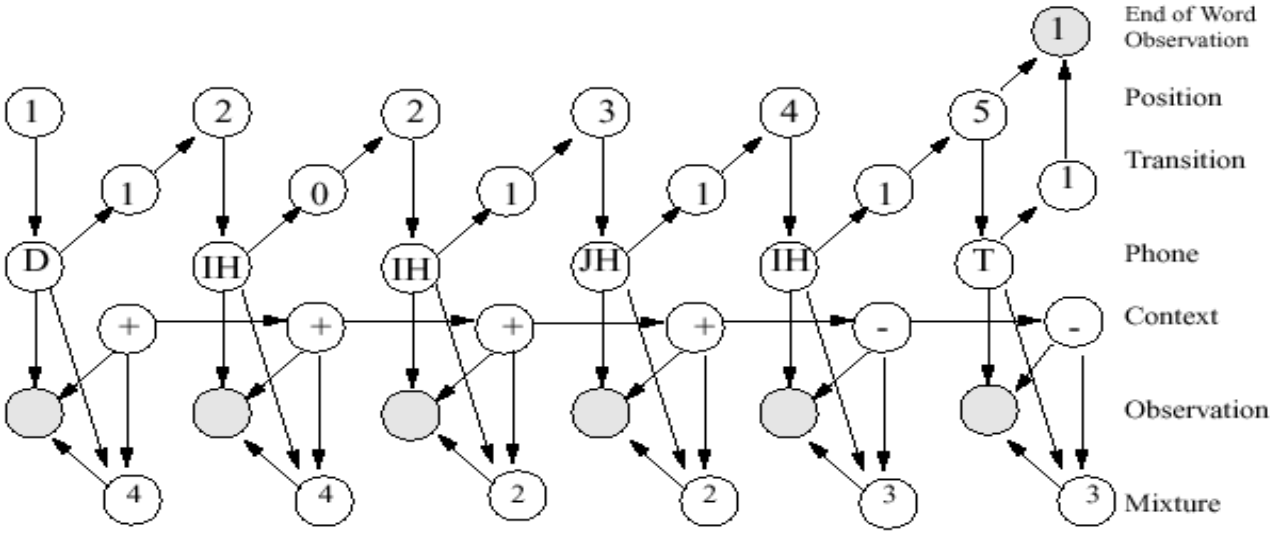

Figure 5. A Bayesian network for speech recognition. The index, transition, phone, and end-of word variables encode a probability distribution over phonetic sequences. The context and observation variables encode a distribution over observations, conditioned on phonetic sequence.

This is the structure shown in Figure 5.

5.3. Performance

The Bayesian networks are tested by using four different model structure on a large-vocabulary, isolated-word recognition task.

- An "articulator" network in which the context variable depends on both the phonetic state and its own past value
- A "chain" network in which the phonetic dependence is removed.
- A "phone-dependent-correlation" network (PD-Correlation) which results from removing the temporal links from the articulator network.
- A "correlation" network which further removes the phonetic dependence.

Network	Parameter	WER
HMM	127k	4.8%
Correlation	254k	3.7%
PD-Correlation	254k	4.2%
Chain	254k	3.6%
Articulator	255k	3.4%

Table 1. Test results with basic phoneme alphabet

Table 1 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. However, most of the differences among the augmented networks are not significant.

6. SUMMARY

In this paper we discussed different methods for constructing Bayesian networks from prior knowledge and how to use Bayesian approach for using given data set to improve the models associated with Bayesian networks. In addition, we demonstrate that Bayesian networks are a flexible tool that can be applied effectively to speech recognition, and show the use of this graphical representation can improve speech recognition results.

7. REFERENCES

[1] Howard. R. and Matheson. J., editors. *The principle and Applications of Decision Analysis*. Strategic Decisions Group, Mento Park, CA. 1983

[2] Tveraky.A. and Kahneman. D. *"Judgment under certainty: Heuristics and biases."* Science. pp. 1124-1131

[3] Spetzler. C. and Stael von Holstein. C. *"Probability encoding in decision analysis"*. Management Science. pp. 340-358

[4] Buntine. W. “Theory refinement on Bayesian networks” In *Proceedings of Seventh Conference on Uncertainty in Artificial Intelligence,* Los Angeles, CA pp. 52-60. Morgan kaufmann.

[5] Geiger. D. and Heckerman. D. “A characterization of the Dirichlet distribution applicable to learning Bayesian networks.” *Technical Report MSR-TR-94-16.* Microsoft Research, Redmond, WA. 1995

[6] Rabiner, L. R. and Juang, B.H. *Fundamentals of Speech Recognition*. Pretice-Hall. 1993

[7] Ghahramani, Z., and Jordan, M.I. Factorial hidden Markov models. *Machine Learning*. 1997.

[8] Smyth. P., Heckerman. D., and Jordan. M. “Probabilistic independence networks for hidden Markov probability models” *Neural Computation* pp. 227-269 1997

[9] Russell, S., et. al., “Local learning in probabilistic networks with hidden variables.” In *IJCAI-95,* pp.1146-1152, Montreal, Canada. 1995

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

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