AUTOMATIC SPEAKER RECOGNITION USING DYNAMIC BAYESIAN NETWORK

Lifeng Sang, Zhaohui Wu, Yingchun Yang, Wanfeng Zhang

Department of Computer Science and Technology,
Zhejiang University, Hangzhou, P.R.China, 310027
{lfsang, wzh, yyc, wfzhang}@cs.zju.edu.cn

ABSTRACT

This paper presents a novel approach to automatic speaker recognition using dynamic Bayesian network (DBN). DBNs have a precise and well-understand probabilistic semantics, and it has the ability to incorporate prior knowledge, to represent arbitrary non-linearities, and to handle hidden variables and missing data in a principled way with high extensibility. Experimental evaluation over YOHO corpus shows promising results compared to other classical methods.

1. INTRODUCTION

Analysis and classification of temporal sequences in automatic speaker recognition has been a focus of research for many years. Many approaches have been developed in this field such as vector quantization (VQ), Gaussian mixture model (GMM), Hidden Markov Model (HMM), that deal with speech and speaker variability to accomplish the task of speaker recognition, but general paradigm is in no way exhausted. Recently, a new statistical approach from the perspective of Bayesian networks was proposed for time series data modeling, as is referred to Dynamic Bayesian Network (DBN). DBNs are knowledge representation schemes that can characterize probability relationships among time series data and perform exact or approximate inference. Zweig [1] first applied DBNs in isolated speech recognition and achieved considerable results. Up to now, DBNs is little used in speaker recognition community. In this paper, we present a novel approach to automatic speaker recognition using dynamic Bayesian network specifically.

As a result of a combination of anatomical differences inherent in the vocal tract and the learned speaking habits, voices of different individuals contain the speaker-related information, and this information can be used to discriminate between speakers. The advantages of using DBNs in speaker recognition lie in two aspects: (1) Time series data of a speaker's voice can be represented by DBNs with high interpretability and flexibility in a unifying statistical framework. (2) Some prior knowledge (e.g. gender, noise) can be described by DBNs conveniently. Our experimental results also show that DBNs is a promising way to modelize the speaker variability.

This paper is organized as the following: as DBNs are not used often in speaker recognition community, we give a brief introduction in section 2. In section 3 and 4, we propose details of inference and learning algorithms in dynamic Bayesian network to the needs of automatic speaker recognition. In section 5, we describe how to recognize a person given his utterances in arithmetic level. Experimental comparison between DBNs and other classical methods such as VQ, GMM, HMM is discussed in section 6. Finally, we give a conclusion in section 7.

2. DYNAMIC BAYESIAN NETWORK

For time-series modeling we can assume that an event can cause another event in the future, but not vice-versa. This simplifies the design of dynamic Bayesian networks allowing directed arcs to flow forward in time.

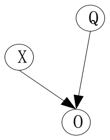

Figure1: A simple Bayesian network

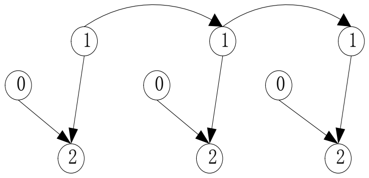

Figure 2: A dynamic Bayesian network with 3 slices

A DBN is a specific type of Bayesian network and is almost always assumed to satisfy the following two conditions: (1) it has the same structure at each time slice *t* and (2) the cross-slice arcs can only be extended from slice *t* to slice *t*+1. Condition (1) means that DBNs are time-invariant so that the topology of the network is a repeating structure, and its conditional probabilities do not change in each time-slice. According to condition (2),

0-7803-7663-3/03/$17.00 ©2003 IEEE

DBNs satisfy the Markov assumption: the future states of the domain are conditionally independent of the past states given the present state. Figure 2 is a simple example of a DBN with 3 slices.

Now, we are going to calculate the joint probability and marginal probability through a simple Bayesian network structure. For simple Bayesian network, as that of figure 1, the joint probability model can be expressed by chain rule:

$$P(Q,X,O)=p(O|X,Q)*P(X|Q)*P(Q) \qquad (1)$$

Since the variable X is independent of variable Q, the joint probability can then be calculated by:

$$P(Q,X,O)=p(O|X,Q)*P(X)*P(Q) \qquad (2)$$

So probability of target $P(O|Q)$ is calculated by marginalization over X:

$$\begin{aligned} P(O|Q) &= \frac{P(O,Q)}{P(Q)}=\frac{\sum_x P(O,Q,X=x)}{P(Q)} \\ &= \frac{\sum_x P(O|X=x,Q)*P(X=x)*P(Q)}{P(Q)} \\ &= \sum_x P(O|X=x,Q)*P(X=x) \end{aligned} \qquad (3)$$

For many practical applications, O can be considered as observation, Q is state variable that drives observation O, and X is other factor variable. The probability $P(O|X=x,Q)$ can be assumed to satisfy Gaussian mixture distributions, and their parameters (means, covariances and weights) can be estimated by the standard EM algorithm from training data.

This is a simple case of Bayesian network. We are going to investigate the complex structure of DBNs and present more general inference algorithm in the following section.

3. INFERENCE IN DBNS

The goal of inference in dynamic Bayesian network is to estimate the posteriori probability of the hidden states of the network given some known sequence of observations O and the known model parameters. Each variable has a probability distribution conditioned on its parent nodes. When a set of observations O is assigned to a subset of the variables in a DBN, the variables left unobserved have their prior probability distribution $P(X_i \mid parent(X_i))$, but need to have their posteriori probability distribution inferred:

$$P(X_i \mid parent(X_i),O) \qquad (4)$$

so the log-likelihood of the observation set $O=\{O_1,O_2,\cdots,O_M\}$ is a sum of terms, one for each node:

$$\begin{aligned} L &= \log\prod_{m=1}^{M}\Pr(O_m \mid G) \\ &= \sum_{i=1}^{N}\sum_{m=1}^{M}\log P(X_i \mid parent(X_i),O_m) \end{aligned} \qquad (5)$$

Here G is a DBN model with N variables.

There are a few of exact and approximate inference algorithms that can be applied to calculate the posteriori probability distributions. One of the most commonly used algorithms is the junction tree algorithm [3], which is similar to the Baum-Welch algorithm used in HMM. Zweig [1] introduced a tailored version to the needs of speech recognition. The junction tree algorithm works with variable λ and π as the following,

$$\lambda_j^i = P(O_i^0, O_i^- \mid X_i = j) \qquad (6)$$

$$\pi_j^i = P(O_i^+, X_i = j) \qquad (7)$$

Here O_i^0 is any observation for X_i itself, O_i^- are the observations for nodes in the subtrees rooted in $X_i^{'}s$ children in the junction tree, O_i^+ are all the remaining observations. Equation (6) and (7) can then be used to compute the marginal probability distribution as well as the joint posteriori probability for each variable.

According to chain rule,

$$\begin{aligned} P(X_i=j,O) &= P(O_i^0,O_i^-,O_i^+,X_i=j) \\ &= P(O_i^+,X_i=j)P(O_i^-,O_i^0 \mid O_i^+,X_i=j) \\ &= P(O_i^+,X_i=j)P(O_i^-,O_i^0 \mid X_i=j) \end{aligned} \qquad (8)$$

So the marginal and joint posteriori probability distribution is calculated as the following,

$$P(X_i=j \mid O)=\frac{\lambda_j^i * \pi_j^i}{\sum_j \lambda_j^i * \pi_j^i}, \quad \forall i \qquad (9)$$

$$P(O)=\sum_j \lambda_j^i * \pi_j^i, \quad \forall i \qquad (10)$$

As we can see, the variables λ and π are analogous to the α and β variables used in HMM respectively. These two variables can be calculated as the following:

1) Computing λ_j^i

If X_i is a leaf node, then

$$\lambda_j^i = 1, \quad \forall\, j \; with \; X_i = j \qquad (11)$$

Otherwise,

$$\lambda_j^i = \prod_{c\in C(X_i)} \sum_f \lambda_f^c * P(X_c = f \mid X_i = j) \quad (12)$$

Here $C(X_i)$ is the set of $X_i^{'}s$ children nodes. Note that to compute a variable's λ, you need to first compute its children's λs.

2) Computing π_j^i

If X_i is the root node, then

$$\pi_j^i = P(X_i = j) \quad (13)$$

Otherwise,

$$\pi_j^i = \sum_v P(X_i = j \mid X_p = v) * \pi_v^p * \prod_{s\in S(X_i)} \sum_f \lambda_f^s * P(X_s = f \mid X_p = v) \quad (14)$$

Here $S(X_i)$ is the set of $X_i^{'}$s siblings nodes. This shows that to compute the value π_j^i, you need to compute its parent's π as well as the conditional probabilities, and its siblings' λs.

4. LEARNING IN DBNS

In our speaker recognition, each speaker is modelized by a DBN model. And all the DBN models are trained independently. Generally, the methods of learning Bayesian network can be divided into 4 types according to the structure and observability of the DBNs [4]: (1) known structure and full observability; (2) known structure and partial observability; (3) unknown structure and full observability; (4) unknown structure and partial observability. Practically, different type of learning method can be applied to different applications under different assumptions. However, this topic is out of the scope of this paper and will be researched in the future. In this speaker recognition, we assume the structure of the DBNs are known for simplicity but have not observed all of the data. In other words, some of the nodes in these DBNs are hidden and some others are observable.

Since discrete DBNs will lose a lot of information, we specify the graph structure and the conditional probability distributions and make it work with continuous variables directly. The most common distribution is a Gaussian, since it is analytically tractable and works successfully in many statistical problems. So for observable node X with a discrete parent node Q (Q is a hidden node in our DBNs), the Gaussian distribution is

$$P(x \mid Q=i) = c\,|\Sigma_i|^{-\frac{1}{2}} \exp(-\frac{1}{2}(x-\mu_i)^{'}\, \Sigma_i^{-1}(x-\mu_i)) \quad (10)$$

Where $c=(2\pi)^{-d/2}$ is a constant and $|x|=d$, μ is the mean vector and $\sum_i$ is the covariance matrix. Given the training sequences, we re-estimate the means and covariance matrices in each iteration using EM algorithm to get the Maximum Likelihood as the following:

$$\mu_i = \frac{\sum_m w_m^i E(x_m \mid Q_m = i, O_m)}{\sum_m w_m^i} \quad (11)$$

$$\Sigma_i = \frac{\sum_m w_m^i E(x_m x_m^{'} \mid Q_m = i, X_m)}{\sum_m w_m^i} - \mu_i \mu_i^{'} \quad (12)$$

Here, function $E(x)$ is the expected likelihood of x, and $w_m^i = E(q_m^i \mid O_m)$. The variable $q_m^i = 1$ if Q has value i in the $m^{'}th$ data cases, and 0 otherwise. See [2] for more general learning techniques in DBNs.

5. HOW TO PERFORM RECOGNITION

Usually, speaker recognition is divided into identification and verification according to its functionality. We will introduce them respectively in the following.

5.1. Identification

The task of identification is to determine if the speaker is a specific one in the group of N known speakers given his utterance. In the closed set problem, it is assured that it belongs to one of the registered speakers. So we need to find the speaker $\hat{i}$ whose DBN model M_i maximizes a posteriori probability $P(M_i \mid O),\ i=1,\cdots,N$. According to Bayes' rule,

$$P(M_i \mid O) = P(O \mid M_i) * P(M_i) / P(O) \quad (13)$$

Since we haven't any prior knowledge of $P(M_i)/P(O)$, we consider it be the same for all speakers. Then the decision rule can be simplified to

$$\hat{i} = \arg\max_i P(O \mid M_i),\ i=1,\cdots,N \quad (14)$$

Here M_i is the DBN model of speaker i, and $\hat{i}$ is the identified speaker. We need to calculate the posteriori probabilities $P(O \mid M_i)$, corresponding to each of the speaker model, and this is can be done using equation (5).

5.2. Verification

The task of verification is to decide whether the speaker is whom he claims to be or not. In many classical approaches to this binary problem, the decision is made by comparing the utterance score of the claimant speaker's model with some prior threshold determined at the training phase. Since the absolute value of utterance score not only represents the speaker's model itself, but also depends on the speech content. Hence a stable threshold can not be set independently. One successful solution is to apply score normalization technique [5].

The decision rule of the verification task is stated as a likelihood ratio given by

$$\frac{\psi(O_i)}{\overline{\psi}(O_i)}\begin{cases} \geq\theta & accept \quad i \\ <\theta & reject \quad i \end{cases} \tag{15}$$

where $\psi(O_i)$ is the probability density function for the hypothesis that observation O_i belongs to the speaker i, while $\overline{\psi}(O_i)$ means that O_i does not belong to the speaker i. The decision threshold for accepting or rejecting is θ. In this speaker recognition, we use background speaker set [5] to deal with the decision rule, so equation (13) can be restated in log domain as

$$\frac{\log P(O_i\,|\,M_i)}{\sum_{j=1,j\neq i}^{N}\log P(O_i\,|\,M_j)}\begin{cases} \geq\theta & accept \quad i \\ <\theta & reject \quad i \end{cases} \quad i=1,\cdots,N \tag{16}$$

6. EXPERIMENTS AND DISCUSSIONS

We use YOHO corpus [6] to evaluate our method in speaker recognition. For computational reasons, only the first enroll session (24 sentences) is used for training and all verify sessions (40 sentences) are used for testing for each speaker. In the feature extraction, the hamming window is 32 mm and the frame shift is 16mm. The silence and unvoiced segments are discarded based on an energy threshold. The feature vectors are composed by 16 MFCC and their delta coefficients.

In our experiments, we define the topology of the DBNs as Figure 3, which is unrolled for first two slices. $q_j^i, i=1,2,3, j=1,2,\cdots T$ are hidden nodes and have discrete values, $o_j^i, i=1,2,3, j=1,2,\cdots T$ can be observed and satisfy Gaussian distributions, here T is the length of time slices.

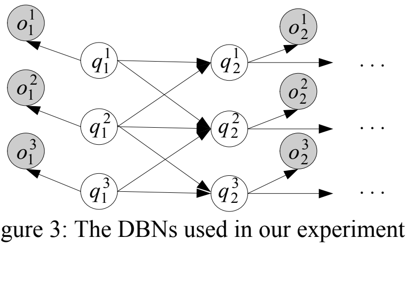

Figure 3: The DBNs used in our experiments

In order to investigate if the method is robust under different sets of different number of speakers, we made experiments on some subsets of YOHO corpus: first 30, first 50 and all 138 speakers. We also compare the DBNs model to other classical methods such as VQ, GMM, HMM. The results are listed in table 1. The considerable performance achieved in the test shows that it is a promising way of using DBNs in speaker recognition.

Method	First 30		First 50		Total 138	
	I (%)	V (%)	I (%)	V (%)	I (%)	V (%)
VQ	88.17	5.79	85.90	5.68	74.18	6.82
GMM	95.25	3.99	94.95	4.03	87.30	5.78
HMM	94.47	4.99	94.20	4.66	86.76	5.87
DBN	96.67	3.00	96.55	3.00	89.93	4.22

Table 1: Experimental results under different speaker number of test sets. I means identification rate, V means equal error rate (EER). In our experiments, the code book size of VQ is 64; The mixture number of GMM is 32. HHM is with 5 states and 10 mixture Gaussian density outputs.

7. CONCLUSIONS

This paper presents an approach of using dynamic Bayesian network in speaker recognition. We discuss how to do inference, learning and testing in DBN for speaker recognition. Encouraging results of experiments on YOHO corpus demonstrate that DBN is a promising way for classification.

This work is supported by National Natural Science Foundation of P.R.China(No.60273059), National High Technology Research & Development Programme (863) of P.R.China (No.2001AA4180), Zhejiang Provincial Natural Science Foundation for Young Scientist of P.R.China (No.RC01058), Zhejiang Provincial Education Office Foundation(20020721), and Zhejiang Provincial Doctoral Subject Foundation (20020335025).

8. REFERENCES

[1]. Zweig, G.G., "Speech Recognition with Dynamic Bayesian Networks. Ph.D. thesis," U.C. Berkeley, 1998

[2]. Murphy, K., "Dynamic Bayesian Networks: Representation, Inference and Learning," Ph.D. thesis, U.C. Berkeley, 2002

[3]. Cowell, R, "Introduction to inference for Bayesian networks," In Jordan, p9-26, 1999

[4]. Murphy, K. and Mian, S., "Modeling gene expression data using dynamic Bayesian networks," Technical Report, U.C. Berkeley, 1999

[5]. Reynolds, D. A., et al., "Speaker verification using adapted Gaussian mixture models," Digital Signal Processing, vol.10, pp. 19-41, 2000

[6]. Campbell, J.Jr., "Testing with the YOHO CD-ROM Voice Verification Corpus," ICASSP 95, pp. 341-345