MIXTURE TREES – HIERARCHICALLY TIED MIXTURE DENSITIES FOR MODELING HMM EMISSION PROBABILITIES

Jürgen Fritsch
fritsch@ira.uka.de

Interactive Systems Labs
University of Karlsruhe, Germany

Interactive Systems Inc.
Pittsburgh, USA

10.21437/Eurospeech.1999-278x

ABSTRACT

We propose a novel hierarchical mixture model and present its application to acoustic modeling for HMM based large vocabulary conversational speech recognition. We detail an EM algorithm for estimating the parameters of such a mixture tree for the case of Gaussian component densities. We sketch how clustering algorithms can be applied to automatically construct suitable mixture trees for a large number of HMM states. Furthermore, we discuss the advantages of a tree structured organization of mixture densities in terms of efficiency of evaluation and scalability of context modeling. These properties allow (1) to arbitrarily downsize trained mixture trees thereby trading off recognition accuracy against decoding speed and model size and (2) to adapt the structure of trained mixture trees in cross domain applications to reflect the differing requirements in specificity of context. We present preliminary results of using mixture trees for recognition experiments on the Switchboard LVCSR corpus.

1. INTRODUCTION

The predominant approach to statistical speech recognition is based on continuous density hidden Markov models (CDHMM), typically using Gaussian mixture densities to model HMM state observation likelihoods. In large vocabulary continuous speech recognition, explicit and robust modeling of phonetic context has proven to be a necessity for achieving state-of-the-art performance, particularly when dealing with spontaneous speech. Typically, phonetic decision trees are applied to cluster context-dependent subphonetic models into so called tied states. Tying is necessary for obtaining robust and well trained models since many of the theoretically possible context-dependent units (e.g. triphones) never occur in actual data.

However, context-dependent modeling comes at the burden of domain dependence. Large domains such as Switchboard and Broadcast News require detailed and highly specific context modeling for optimal performance. In contrast, smaller domains such as digit recognition tasks show much smaller specificity of phonetic context and require only a moderate amount of context modeling. Furthermore, while the a-priori distribution of context-independent phones appears to be relatively constant over many domains, context-dependent phone models exhibit vast differences in a-priori distribution for various domains [2].

A solution to the above problem requires scalable specificity of context modeling which can be achieved by hierarchically structured acoustic models. We have demonstrated the benefits of such an approach in previous work [2, 3, 4] where we have applied conditional factoring of posterior state probabilities to devise a hierarchical architecture of connectionist estimators for scalable context-dependent acoustic modeling.

In this paper, we present mixture trees, a novel but related hierarchical model that, while estimating state likelihoods instead of state posteriors, offers comparable scalability and structural advantages as the model presented in [4]. Mixture trees can be regarded as collections of mixture densities with tree-structured tying of component densities. Tree nodes represent (shared) component densities and paths from root to the leaves correspond to individual mixture densities. As we will see, mixture trees allow to represent coarse to fine grain context models within a single tree structure. This multi-level modeling and representation of phonetic context through a tree structure yields the following advantages:

- Acoustic models based on mixture trees can be ported effectively to previously unseen domains by adapting and pruning the tree structure to reflect the differing requirements in specificity of context.
- Mixture trees already represent the structure required to deal with limited amounts of adaptation data as is typically the case in speaker adaptation. There is no need for additional structures such as regression trees often used with conventional models since parameter sharing is inherently realized.
- Without having to re-estimate a single parameter, trained mixture trees can be downsized arbitrarily to model a smaller number of HMM states, thereby reducing model size and decoding speed, possibly at the cost of increased word error rate.

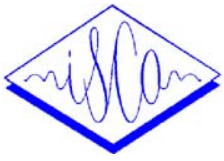
Although there are strategies for downsizing conventional acoustic models (e.g. [6]) by exploiting the structure of phonetic decision trees, such approaches lack true scalability as they require to compute a set of new mixture densities from the original ones for each downsized tree. The idea of hierarchically structuring probability densities has already been expressed in [5] and [8]. However, that work focused on speeding up the evaluation of acoustic models and used an additional tree structure only for quickly selecting a subset of the original, unstructured densities to be evaluated. In contrast, the present approach consists of a recursively defined, tree-structured mixture model that successively refines estimates of state likelihoods from the root to the leaves of the tree, which correspond to HMM states.

2. HIERARCHICALLY TIED MIXTURE DENSITIES

We consider the task of estimating HMM state observation likelihoods for a set of N decision tree clustered states s_i using mixture densities. In a conventional CDHMM setting, we model each state *independently* according to

$$p_i(\mathbf{x}) = p(\mathbf{x}|s_i) = \sum_{k=1}^{K_i} c_i^{(k)}\, q_i^{(k)}(\mathbf{x}) \qquad \forall i \in \{1, \ldots, N\}$$

where the $c_i^{(k)}$ are (affine) mixture weights satisfying $\sum_{k=1}^{K_i} c_i^{(k)} = 1$ and $c_i^{(k)} \geq 0$, and the $q_i^{(k)}(\mathbf{x})$ are mixture component densities in the space of feature vectors $\mathbf{x}$. Mixture densities are usually prefered over simple densities because of their universal approximation property.



6th European Conference on
Speech Communication and Technology
(EUROSPEECH'99)
Budapest, Hungary, September 5-9, 1999

2.1. Mixture Trees

Mixture trees are motivated by the observation that individual mixture densities of context-dependent speech models overlap considerably in feature space. Instead of using separate sets of component densities for each mixture density, we can share some of the component densities to allow for joint modeling of the overlapping parts of the distributions. For instance, consider the set of 4 mixture densities depicted in the upper plot of Fig. 1. Each one of the densities consists of 3 vertically organized component densities. Instead of assigning 3 component densities exclusively to each mixture, we share some of them between adjacent mixtures such that one component is used exclusively, one is shared between two and another one is shared between all four densities. We call the resulting tree structured configuration (Fig. 1) of component densities a *mixture tree*. The equivalent to mixture weights in standard mixtures are interpolation weights between tree nodes.

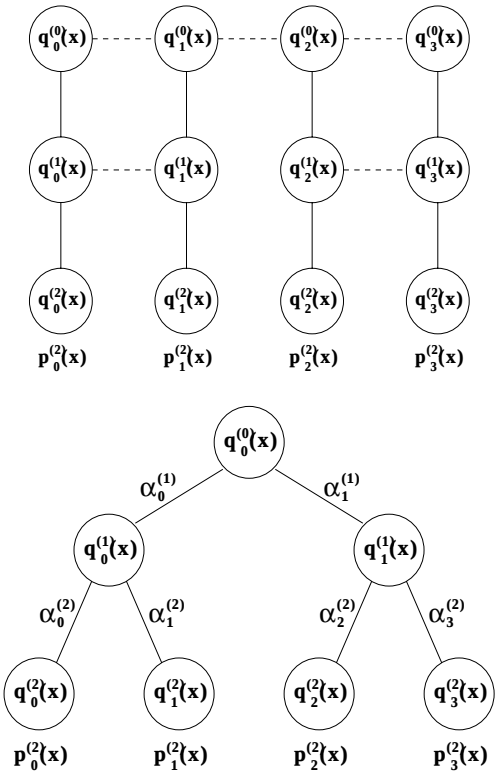


Figure 1. Hierarchically Tied Mixture Densities ($d = 2, b = 2$)

Introducing depth d and branching factor b of a mixture tree, we rewrite the state observation likelihoods, now being estimated by the leaves of the mixture tree, as $p_i(\mathbf{x}) = p_i^{(d)}(\mathbf{x})$ and recursively define the mixture model as

$$\begin{aligned} p_0^{(0)}(\mathbf{x}) &= q_0^{(0)}(\mathbf{x}) \\ p_i^{(k)}(\mathbf{x}) &= \alpha_i^{(k)} q_i^{(k)}(\mathbf{x}) \; + \; (1 - \alpha_i^{(k)}) \; p_{\lfloor i/b \rfloor}^{(k-1)}(\mathbf{x}) \end{aligned}$$

where the $q_i^{(k)}$ are (tied) component densities and the $\alpha_i^{(k)}$ are local mixture weights, satisfying $0 \leq \alpha_i^{(k)} \leq 1$. An individual mixture density represented in the tree is evaluated top-down, starting at the root node. Proceeding down the tree towards the corresponding leaf node, we incrementally refine the current estimate by computing affine interpolations between the already accumulated partial mixture probability and the current local component density using weights $\alpha_i^{(k)}$. An interesting aspect of this model is that any node in the mixture tree (not just the leaf nodes) computes a valid probability density which depends only on predecessor nodes. This property allows to downsize the mixture tree without having to re-estimate any of the parameters. In fact, the partial mixture probability computed down to a specific node represents the probability of the feature vector being generated by any of the leaf nodes (states) in the corresponding subtree. This implies that the root component density $q_0^{(0)}(\mathbf{x})$ models the unconditional density $p(\mathbf{x})$ of the data.

2.2. Parameter Estimation (Forward Backward)

In a maximum likelihood framework, the parameters of a mixture model can be estimated iteratively using an Expectation - Maximization (EM) algorithm [1, 7]. If the forward-backward (Baum-Welch) algorithm is used for training the HMMs, we face two nested probabilistic models; (1) assigning HMM states to observations and (2) assigning mixture component densities within state mixtures to observations.

We give an EM algorithm for mixture trees consisting of Gaussian component densities. The resulting algorithm can easily be generalized to other types of component densities. First we have to compute posteriors $h_i^{(k)}(\mathbf{x})$ for all nodes in the mixture tree in a top-down fashion given the current set of parameters. This constitutes the E-step of our EM algorithm:

$$\begin{aligned} h_0^{(0)}(\mathbf{x}) &= 1 \\ h_i^{(k)}(\mathbf{x}) &= \frac{\alpha_i^{(k)} q_i^{(k)}(\mathbf{x})}{p_i^{(k)}(\mathbf{x})} \end{aligned}$$

Here, $h_i^{(k)}(\mathbf{x})$ measures the posterior probability of component $q_i^{(k)}$ contributing to the mixture down to the current depth, given the current feature vector. Again, it is important to note that the node posteriors in our model depend only on parent nodes, not on any of the child nodes in the tree.

In the M-step, we update the mixture weights $\alpha_i^{(k)}$ and the parameters of the component densities $q_i^{(k)}$ based on the expectations for all training patterns gained in the E-step such that the likelihood of the model given the data increases. We obtain the following updates for the node parameters:

$$\begin{aligned} \alpha_i^{(k)} &= \frac{\sum_{m=1}^{M} \gamma_i^{(k)}(\mathbf{x}_m) \; h_i^{(k)}(\mathbf{x}_m)}{\sum_{m=1}^{M} \gamma_i^{(k)}(\mathbf{x}_m)} \\ \mu_i^{(k)} &= \frac{\sum_{m=1}^{M} \gamma_i^{(k)}(\mathbf{x}_m) \; h_i^{(k)}(\mathbf{x}_m) \; \mathbf{x}_m}{\sum_{m=1}^{M} \gamma_i^{(k)}(\mathbf{x}_m) \; h_i^{(k)}(\mathbf{x}_m)} \\ \mathbf{\Sigma}_i^{(k)} &= \frac{\sum_{m=1}^{M} \gamma_i^{(k)}(\mathbf{x}_m) \; h_i^{(k)}(\mathbf{x}_m) \; \mathbf{x}_m \mathbf{x}_m^t}{\sum_{m=1}^{M} \gamma_i^{(k)}(\mathbf{x}_m) \; h_i^{(k)}(\mathbf{x}_m)} \end{aligned}$$

where $\mu_i^{(k)}$ is the mean vector and $\mathbf{\Sigma}_i^{(k)}$ is the covariance matrix of the Gaussian density $q_i^{(k)}$. The weights $\gamma_i^{(k)}(\mathbf{x})$ (called node occupation probabilities) are computed in a bottom-up fashion from the state occupation probabilities $\gamma_i(\mathbf{x})$ (obtained in the forward-backward algorithm) according to

$$\begin{aligned} \gamma_i^{(d)}(\mathbf{x}) &= \gamma_i(\mathbf{x}) \\ \gamma_i^{(k)}(\mathbf{x}) &= \sum_{j=bi}^{b(i+1)-1} \gamma_j^{(k+1)}(\mathbf{x}) \end{aligned}$$

That is, a node occupation probability is computed as the sum of all state occupation probabilities of all states (leaves) in the corresponding subtree.

2.3. Parameter Estimation (Viterbi)

In the case of Viterbi training, a state alignment implies a one-to-one mapping between HMM states and feature vectors. Thus, for any input feature vector, there is exactly one state with state occupation probability $\gamma_i(\mathbf{x}) = 1$, all other state occupation probabilities vanish. In a mixture tree, the Viterbi assumption leads to a single path of non-zero node occupation probabilities $\gamma_i^{(k)}(\mathbf{x}) = 1$ from root to one of the leaves for each feature vector. Therefore, node posteriors in the E-step have to be evaluated only along the path through the mixture tree that corresponds to the current pair of feature vector and HMM state.

2.4. Initialization

As with standard mixture densities, reasonable initialization of parameters is crucial for rapid convergence of the EM algorithm. In the case of hierarchically tied mixture densities with Gaussian component densities, we initialize local mixture weights according to

$$\alpha_i^{(k)} = \frac{1}{k+1}$$

which corresponds to a uniform component prior distribution for all mixtures in the mixture tree. Individual Gaussian component densities are initialized using the ML estimates for the forward-backward weighted data observed at the corresponding tree node:

$$\begin{aligned} \boldsymbol{\mu}_i^{(k)} &= \frac{\sum_{m=1}^{M} \gamma_i^{(k)}(\mathbf{x}_m)\ \mathbf{x}_m}{\sum_{m=1}^{M} \gamma_i^{(k)}(\mathbf{x}_m)} \\ \boldsymbol{\Sigma}_i^{(k)} &= \frac{\sum_{m=1}^{M} \gamma_i^{(k)}(\mathbf{x}_m)\ \mathbf{x}_m \mathbf{x}_m^t}{\sum_{m=1}^{M} \gamma_i^{(k)}(\mathbf{x}_m)} \end{aligned}$$

In case of Viterbi training, this simplifies to the ML estimates for all data of all states (leaf nodes) found in the subtree of the node to be initialized.

2.5. Constructing Mixture Trees

Up to now, we have assumed the availability of a suitable tree structure. The set of context clustering decision trees constitutes such a structure and it is in principle possible to adopt this structure for hierarchical acoustic modeling. However, for the experiments reported in this paper, we have chosen to use decision trees only for clustering phonetic contexts into a set of tied states and to construct a separate tree structure suitable for mixture tree modeling of these states. This way,

- acoustic similarities between allophones of different monophones can be taken into account,
- the mixture tree can be forced to be balanced such that each mixture in the tree consists of approximately the same number of component densities.

We have experimented with both agglomerative and divisive clustering algorithms based on measures of acoustic similarity between HMM states to automatically construct a tree structure for the large number of (tied) HMM states typically used in LVCSR systems. For computing appropriate measures of similarity, we first estimate simple Gaussian densities for each state to be modeled by the mixture tree. Various measures of similarity can be defined for states modeled by Gaussian densities, for instance information divergence or likelihood gain.
In [4], we presented an agglomerative clustering algorithm for constructing a tree structure based on information divergence. Although this algorithm was developed in the context of connectionist hierarchical acoustic modeling it can just as well be applied to construct mixture trees. Alternatively, a locally greedy divisive clustering algorithm can be devised which is computationally more efficient and allows to construct trees of arbitrary branching factor b. We used split likelihood gain for divisive clustering. Starting with the root node, we successively split nodes by partitioning the contained set of Gaussians such that the gain in likelihood is maximized, until we reach leaf nodes, containing just a single Gaussian. Since the investigation of all $b^n - b(b-1)^n$ legal partitionings of a set of n Gaussians at a node with branching factor b is intractable, we apply a greedy optimization algorithm that, starting from some random initial configuration, iteratively tries to improve the split likelihood gain by relocating individual Gaussians from one branch into another.

3. PRUNING MIXTURE TREES

As already mentioned, each node in a mixture tree estimates the likelihood of the corresponding subtree, with the leaves representing (tied) HMM states. In order to compute the likelihood of a specific state, we have to follow the path from the root node to the leaf corresponding to that state, refining estimates of the likelihood at each node. Instead of traversing the tree all the way down to the leaves, we can stop computing refined likelihoods at any tree level and treat all states in the remaining subtree as a new tied state. This way, the specificity of context-dependent modeling and the number of distinctly modeled HMM states of a trained mixture tree can be reduced arbitrarily, from full context-dependent modeling down to context-independent modeling and further. Introducing pruning depth d_p, we can remove all tree nodes at depths $d > d_p$ and consider all nodes at depth $d = d_p$ to be the new set of distinct HMM states, thereby reducing both the size of the mixture tree and the amount of computations required to evaluate the state likelihoods. Thus, pruning of mixture trees allows to easily adapt recognizers to available memory and/or processor speed without having to re-train or re-cluster the system.

4. EXPERIMENTS

For our preliminary experiments, we chose the Switchboard large vocabulary conversational speech corpus. We used an acoustic frontend that computes 42 dimensional feature vectors consisting of 13 mel-frequency cepstral coefficients plus log power and their first and second derivatives. To compensate speaker and channel differences, we also applied speaker-dependent frequency warping (VTLN) and cepstral mean and variance normalization. Quinphone HMMs with a total of 8000 tied states were constructed from Switchboard and Callhome data by building about 150 phonetic decision trees, one for each state of context-independent 3 state HMM models. Top-down decision tree clustering was based on split likelihood gain using diagonal Gaussians to model state distributions.
We then used divisive clustering to construct a binary mixture tree for the 8000 states. Non-uniform priors were penalized during tree construction in order to obtain a balanced tree. The final mixture tree had a maximum depth of 18. Simple diagonal Gaussians were chosen as component densities in each node. After initialization according to section 2.4, we trained the mixture tree for 4 iterations using Viterbi state alignments of 170 hours of Switchboard data from a conventional recognizer. To improve modeling accuracy, we then replaced the Gaussian component densities in each tree node by mixtures of 8 Gaussians that were obtained from the original Gaussian by translating means randomly according to variances (mixing-up). The resulting mixture tree, containing a total of 127992 Gaussians in 15999 nodes, was trained for another 6 iterations, until training data likelihood converged.
Fig. 2 depicts mean and standard deviation of interpolation weights α in each level of the trained mixture tree. For increasing tree depth, interpolation weights get smaller consistent with the initialization strategy and eventually level off at a mean of around 0.3. Their variance increases slightly towards the bottom of the tree which might indicate saturation of the specificity of context-modeling in some branches of the tree.
Next, we evaluated the performance of the trained mixture tree in recognition experiments. All recognition runs used a

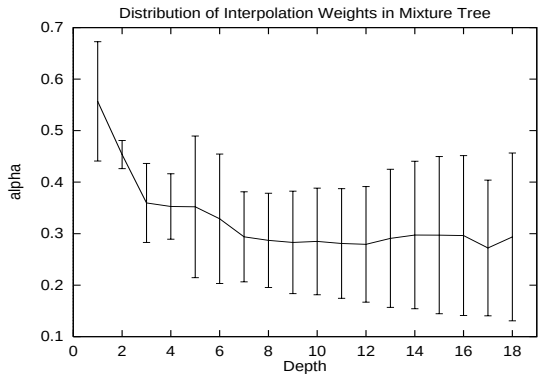


Figure 2. Distribution of Interpolation Weights in Mixture Tree

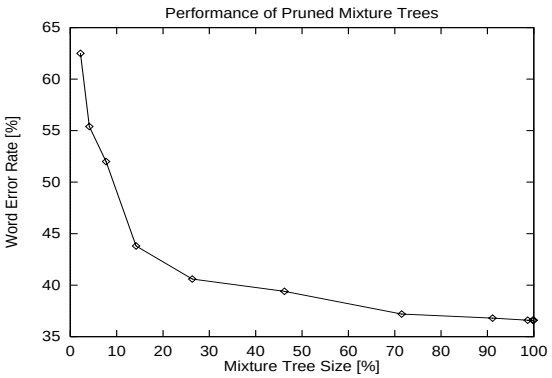


Figure 3. Word Error Rate vs. Tree Size

15k vocabulary and a standard trigram language model trained from the 3.5 million words in the Switchboard corpus. The results reported here were obtained on a test set consisting of the first 30 secs from 12 representative speakers taken from the 1997 development test set and contained a total of 1340 words. Using the full mixture tree with mixtures of 8 Gaussians as component densities in each node, we achieved an unadapted word error rate of 36.6% on this test set. For comparison, we trained and tested a conventional model based on mixtures of Gaussians on the same data. In order to compare models with approximately equal number of parameters, we used mixtures of 16 Gaussians for each of the 8000 states. Using the conventional model with the same recognition setup, we achieved a comparable, slightly better unadapted word error rate of 36.1% on the above test set.

Finally, we investigated the effects of pruning the mixture tree. The original tree of depth 18 that models 8000 tied states was successively pruned with d_p ranging from 17 down to 8, where d_p=8 resulted in a tree with only 179 distinctly modeled states, corresponding roughly to the number of states in a context-independent system. We decoded the above test set for each pruned mixture tree, using the exact same decoder parameters as with the unpruned mixture tree. Table 1 summarizes the results obtained with the original and the pruned trees regarding size, overall decoding speed and word error rate on the above test set.

pruning depth	# tied states	# nodes in tree	tree size [%]	speed [xRT]	WER [%]
–	8000	15999	100.0	48	36.6
17	7991	15983	99.8	48	36.6
16	7897	15795	98.7	48	36.6
15	7290	14581	91.1	45	36.7
14	5722	11445	71.5	39	37.2
13	3699	7399	46.2	31	39.4
12	2109	4219	26.3	24	40.6
11	1143	2287	14.2	19	43.8
10	619	1239	7.7	17	52.0
9	331	663	4.1	16	55.4
8	179	359	2.2	16	62.5

Table 1. Summary of Results with Mixture Trees

While the speed-up in evaluating likelihoods (not shown in Table 1) that can be achieved by pruning the mixture tree corresponds roughly to the reduction in tree size, the speed-up for overall recognition time depends on the proportion of time spent in actual decoding which can significantly exceed the proportion of time spent in evaluation of acoustic likelihoods. The highest speed-ups can be expected for close to realtime systems. In our case, decoding with the smallest tree was about three times faster than decoding with the full tree.

Fig. 3 depicts a plot of word error rate vs. mixture tree size for the results summarized in Table 1. As expected, the performance for the smallest tree, modeling 179 distinct HMM states is comparable to what is typically reported for context-independent Switchboard systems. On the other hand, the mixture tree can be pruned to about 25% of its orginal size at the cost of only moderate increases in word error rate of about 4% absolute.

5. CONCLUSIONS

We have presented mixture trees, a hierarchically structured architecture for the estimation of HMM state observation likelihoods. The paper details model evaluation, parameter estimation and tree construction for Gaussian based mixture trees. Based on a multi-level tree-structured tying of component densities, our approach allows to represent coarse to fine grain context models within a single acoustic model. Exploiting this property, mixture trees can be downsized arbitrarily, trading-off model size and decoding speed against recognition accuracy. Preliminary experiments with the proposed architecture on Switchboard demonstrate the viability of the proposed model and pruning method.

ACKNOWLEDGMENTS

This research greatly benefited from many fruitful discussions and collaborative work with Michael Finke and Detlef Koll.

REFERENCES

[1] A. P. Dempster, N. M. Laird, D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm", In *Journal of the Royal Statistical Society, B 39(1)*, 1977.

[2] J. Fritsch, M. Finke, A. Waibel, "Effective Structural Adaptation of LVCSR Systems to Unseen Domains Using Hierarchical Connectionist Acoustic Models", In *Proc. of ICSLP'98*, Sydney 1998.

[3] J. Fritsch, M. Finke, "ACID/HNN: Clustering Hierarchies of Neural Networks for Context-Dependent Connectionist Acoustic Modeling", In *Proc. of ICASSP'98*, Seattle, Wa, 1998.

[4] J. Fritsch, "ACID/HNN: A Framework for Hierarchical Connectionist Acoustic Modeling", In *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding*, Santa Barbara, 1997.

[5] J. Fritsch, I. Rogina, "The Bucket Box Intersection (BBI) Algorithm for Fast Approximative Evaluation of Diagonal Mixture Gaussians", *Proc. of ICASSP'96*, Atlanta 1996.

[6] M. Hwang, X. Huang, "Dynamically Configurable Acoustic Models for Speech Recognition", *Proc. of ICASSP'98*, Seattle 1998.

[7] R. A. Redner, H. F. Walker, "Mixture Densities, Maximum Likelihood and the EM Algorithm", In *SIAM Review 26*, 1984.

[8] T. Watanabe, K. Shinoda, K. Takagi, E. Yamada, "Speech Recognition using Tree-Structured Probability Density Functions", in Proc. of ICSLP'94, Yokohama, 1994.