EURASIP Journal on Audio, Speech, and Music Processing
a SpringerOpen Journal

RESEARCH Open Access

Decision tree-based acoustic models for speech recognition

Masami Akamine[1*] and Jitendra Ajmera[2]

Abstract

This article proposes a new acoustic model using decision trees (DTs) as replacements for Gaussian mixture models (GMM) to compute the observation likelihoods for a given hidden Markov model state in a speech recognition system. DTs have a number of advantageous properties, such as that they do not impose restrictions on the number or types of features, and that they automatically perform feature selection. This article explores and exploits DTs for the purpose of large vocabulary speech recognition. Equal and decoding questions have newly been introduced into DTs to directly model gender- and context-dependent acoustic space. Experimental results for the 5k ARPA wall-street-journal task show that context information significantly improves the performance of DT-based acoustic models as expected. Context-dependent DT-based models are highly compact compared to conventional GMM-based acoustic models. This means that the proposed models have effective data-sharing across various context classes.

Keywords: speech recognition, acoustic modeling, decision trees, probability estimation, likelihood computation

1. Introduction

Gaussian mixture models (GMMs) are commonly used in state-of-the-art speech recognizers based on hidden Markov models (HMMs) to model the state probability density functions (PDFs) [1]. These state PDFs estimate the likelihood of a speech sample, $\mathbf{X}$, given a particular *state* of the HMM, denoted as $P(\mathbf{X}|s)$. The sample $\mathbf{X}$ is typically a vector representing the speech signal over a short time window, e.g., Mel frequency cepstral coefficients (MFCCs). Recently, some attempts have been made to use decision trees (DTs) for computing the acoustic state likelihoods instead of GMMs [2-6].[a]

While DTs are powerful statistical tools and have widely been used for many pattern recognition applications, their effective usage in ASR has mostly been limited to state-tying prior to building context-dependent acoustic models [7]. In DT-based acoustic modeling, DTs are used to determine the state likelihood by asking a series of questions about the current speech observation. Starting from the root node of the tree, appropriate questions are asked at each level. Based on the answer to the question, an appropriate child node is selected and evaluated next. This process is repeated until the selected node is a leaf node, which provides the pre-computed likelihood of the observation given the HMM state. The question at each node can involve a scalar or a vector value.

In [2], Foote treated DTs as an improvement of vector quantization in discrete acoustic models and proposed a training method for binary trees with hard decisions. We view a DT in [3,5] as a tree-based model with an integrated decision-making component. In [5], we proposed soft DTs to improve robustness against noise or any mismatch in feature statistics between training and recognition. Droppo et al. [4] explored DTs with vector-valued questions. However, in each of these, only simple tasks such as digit or phoneme recognition have been explored.

DTs are attractive for a number of reasons including their simplicity, interpretability, and ability to better incorporate categorical information. If used as acoustic models, they can offer additional advantages over GMMs: they make no assumptions about the distribution of underlying data; they can use information from many different sources, ranging from low-level acoustic features to high-level information such as gender,

* Correspondence: masa.akamine@toshiba.co.jp
[1]Toshiba Corporate R&D Center, 1, Komukai Toshiba, Saiwai, Kawasaki 212-8582, Japan
Full list of author information is available at the end of the article

phonetic contexts, and acoustic environments; and they are computationally very simple. Prior to this article these advantages have not fully been explored.

This article explores and exploits DTs for the purpose of large vocabulary speech recognition [7]. We propose various methods to improve DT-based acoustic models (DTAMs). In addition to the continuous acoustic feature questions previously asked in the DTAMs, the use of discrete category matching questions (e.g., gender = male), and decoding state-dependent phonetic context questions are investigated. We present various configurations of a DT forest, i.e., a mixture of DTs and their training.

The remainder of this article is organized as follows. Section 2 presents an overview of the proposed acoustic models including model training. Section 3 introduces equal and decoding questions and Section 4 presents various ways of realizing the forest. Section 5 presents the experimental framework and evaluation of various proposed configurations. Finally, Section 6 concludes this article.

2. DT-based acoustic models

As shown in Figure 1, DTAMs are HMM-based acoustic models that utilize DTs instead of GMMs to compute observation likelihoods. A DT determines the likelihood of an observation by asking a series of questions about the current observation. Questions are asked at question nodes, starting at the root node of the tree, ending at a leaf node that contains the pre-computed likelihood of the observation given the HMM state.

Throughout this article, we assume that DTs are implemented as binary trees. DTs can deal with multiple target classes at the same time [8] and this makes it possible to use a single DT for all HMM states [4]. However, we found from preliminary experiments that better results are obtained by using a different tree for each HMM state of a context-independent model set. We deal with only hard decisions in this article whereas we proposed soft decisions in [5]. It is straightforward to extend the methods presented in this article to soft decisions. At each node, questions are asked about the observed acoustic features of the form, for example, $x_j \leq s_d$? where x_j is the jth element of the observed acoustic feature vector $\mathbf{X}$, with numerical values, and s_d is the corresponding threshold. This type of question is referred to as an *acoustic* (*numerical*) *question*.

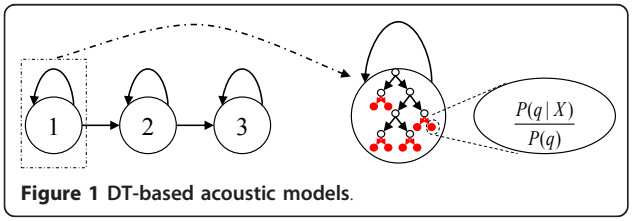

Figure 1 DT-based acoustic models.

Each DT is trained to discriminate between the training data that correspond to the associated HMM state ("true" samples) and all other data ("false" samples). The scaled likelihood of the D-dimensional observation $\boldsymbol{X} = (x_1, x_2, ..., x_j, ..., x_D)$ given state q can then be computed using:

$$L(\boldsymbol{X}|q) = \frac{P(q|\boldsymbol{X}) \cdot P(\boldsymbol{X})}{P(q)} \quad (1)$$

where $P(q|\boldsymbol{X})$ is the posterior probability of state q given observation $\boldsymbol{X}$, $P(q)$ is the prior probability of state q, and $P(\boldsymbol{X})$ is the probability of observation. $P(\boldsymbol{X})$ is independent from the questions asked in the DT and is ignored in training and decoding. The likelihood given by the above equation is stored in each leaf node.

The parameter estimation process for the DTs consists of a growing stage, followed by an optional bottom-up pruning stage. A binary DT is grown by splitting a node into two child nodes as shown in Figure 2. The training algorithm considers all possible splits, i.e., evaluating every feature and corresponding threshold, and selects the split that maximizes the split criterion and meets a number of other requirements. Specifically, splits must pass a chi-square test and must result in leaves with a sufficiently large number of samples. This helps us avoid problems with over-fitting. For this article, the split criterion used was the total log likelihood increase of the true samples. Other criteria such as entropy impurity or Gini impurity can be used. There are two reasons why we use the likelihood gain: (1) Since the log likelihood values are used in a generative model like a HMM, it is a better choice to optimize the split based on the same criterion as that HMMs use; (2) As explained later

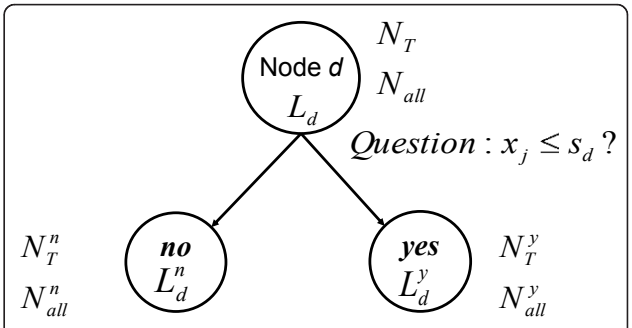

Figure 2 A split based on a question that maximizes the increase in log likelihood. N_T and N_{all} are the number of true samples reaching node d and the total number of samples (true and false) assigned to node d. N_T^y and N_{all}^y are the numbers of the true and all samples at child yes, N_T^n and N_{all}^n are the numbers of the true and all samples at child no, respectively

(Section 3), DTAMs can use not only acoustic questions but also decoding questions. Consistent use of both types of questions requires a criterion that can incorporate prior probabilities. This is not the case with entropy impurity and Gini impurity.

If the number of true samples reaching a node (node d) is N_T and the total number of samples (true and false) is N_{all}, the likelihood at node d, L_d is given by

$$P\left(\boldsymbol{X}|q\right) \propto \frac{P\left(q|X\right)}{P\left(q\right)} = \frac{N_T}{N_{all} \cdot p} = L_d \tag{2}$$

where $p = P(q)$ is the *prior* probability of *state* q and is given by the frequency of the samples assigned to the root node out of all the training set samples. Therefore, the increase of the total log likelihood ΔL from the split is

$$\Delta L = N_T^y log L_d^y + N_T^n log L_d^n - N_T log L_d \tag{3}$$

$$L_d = \frac{N_T}{N_{all} \cdot p}, \quad L_d^y = \frac{N_T^y}{N_{all}^y \cdot p}, \quad L_d^n = \frac{N_T^n}{N_{all}^n \cdot p} \tag{4}$$

where L_d, L_d^y, and L_d^n are the likelihoods at node d, at the child node of node d answering the split question with *yes* (denoted "child *yes*"), and at the other child node answering with *no* (denoted "child *no*"), respectively. Where N_T^y and N_{all}^y are the numbers of the true and all samples at child *yes*, N_T^n and are the numbers of the true and all samples at child *no*, respectively, as shown in Figure 2. N_{all}^y and N_{all}^n samples are propagated to further nodes from the child node yes and the child node no, respectively.

Since we are dealing with one scalar component of the representation at a time, for each node it is possible to perform an exhaustive search over all possible values of x_j and s_d to find the best question that maximizes ΔL in Equation (2). Alternatively, the sample mean of data arriving at a node can be used to set the threshold value s_d. Thus, we obtain the best value of the threshold and the corresponding feature component in the feature vector for one node at a time, and then move down to the next node.

The process of splitting is continued as long as there are nodes which meet the above-mentioned conditions. When a node cannot be split any further, it is referred to as a leaf node and its leaf-value provides the likelihood of sample $\boldsymbol{X}$ given by Equation (2) where N_T^l and N_{all}^l are the numbers of the true and all samples at the leaf node, l, respectively.

Once a tree is fully grown, the DT can be pruned in a bottom-up fashion to improve the robustness of the likelihood estimates for unseen data and to avoid overfitting. The likelihood split criterion can be used to prune the tree. We apply the bottom-up pruning to the tree using development data, held out from the training data set, as for context clustering in conventional GMM based systems, i.e., worst-first fashion. This pruning can also be applied to keep the number of parameters in the proposed DTAM systems comparable to a GMM-based baseline system for comparison purposes.

After the initial DTs are constructed from the training alignments, the HMM transition parameters and DT leaf values are re-estimated using several iterations of the Baum-Welch algorithm [1]. Depending on the quality of the initial alignments, the process of growing trees and re-estimating the parameters can be repeated until a desired stopping criterion has been reached, such as a maximum number of iterations. The full steps for growing the DTs and training the DTAMs are as follows:

1. Generate state-level alignments on the training data set using a bootstrap model set.
2. Grow DTs and generate initial DTAMs.
3. Optionally perform bottom up pruning on a held-out development data set.
4. Generate new state-level alignments for the training data set using Viterbi decoding with the most recent DTAMs.
5. Re-estimate the leaf values and HMM transition parameters based on the alignments from four and most recent DTAMs.
6. Iterate steps 4-6 until desired stopping criterion reached.

3. Integration of high-level information

One of the biggest potential advantages of DTAMs over GMMs is that they can efficiently embed unordered or categorical information such as gender, channel, and phonetic context within the core model. This means that training data that does not vary much over different contexts can be shared instead of having to split at a very high level such as gender dependent GMM-based HMMs. A question in the form $a = Type$? is used for this purpose where a is one of the attributes (e.g., gender) of the data. There are two cases where these questions are implemented. One is where the questions are independent of decoding states and can be treated in the same manner as acoustic questions except asking if the attribute equals a specific type. This type of question is referred to as an *equal question*. The other is where the questions are dependent on decoding states and are treated differently. This type is referred to as a *decoding question*.

3.1. Equal questions

This type of question can be asked in the same manner as the acoustic questions described in Section 2. In this

case, the corresponding leaf-values represent $P(q|\boldsymbol{X}, a = Type)/P(q)$ and the following equation stands:

$$\frac{P\left(q|\boldsymbol{X}, a = Type\right)}{P\left(q\right)} \propto \frac{P\left(\boldsymbol{X}, a = Type|q\right)}{P\left(a = Type|\boldsymbol{X}\right)}. \tag{5}$$

Therefore, the left-hand side of Equation (5) is proportional to the likelihood. The log likelihood is computed at a child node according to the answer to the question $a = Type$?:

$$L = \begin{cases} log\dfrac{N_T^y}{N_{all}^y \cdot p}; \text{at child } yes \\ log\dfrac{N_T^n}{N_{all}^n \cdot p}; \text{at child } no. \end{cases} \tag{6}$$

The overall log likelihood can be computed as a weighted sum of the log likelihood at each child:

$$L = N_T^y log\frac{N_T^y}{N_{all}^y \cdot p} + N_T^n log\frac{N_T^n}{N_{all}^n \cdot p} \tag{7}$$

where N_T^y and N_{all}^y are the numbers of the true and all samples at child *yes*, N_T^n and N_{all}^n are the numbers of the true and all samples at child *no*, respectively. p is the prior probability of state q.

This is applicable for information such as gender. At the time of training when the gender information is available, the overall log likelihood at each node is computed using Equation (7) and the best split is found in the same manner as the acoustic questions. Unlike the acoustic feature data used previously, the categorical information may not be available at decoding time. In this case, the information will have to be predicted. For example, if the gender information is provided at decoding, the log likelihood is given by Equation (6). However, if the gender information is probabilistically computed as $P(gender = male/female|\boldsymbol{X})$ after the test data sample $\boldsymbol{X}$ is observed, the log likelihood can be computed as a weighted sum of those at child nodes:

$$L = P\left(male|\boldsymbol{X}\right) \cdot L^y + P\left(female|\boldsymbol{X}\right) \cdot L^n \tag{8}$$

where, L^y and L^n are the log likelihoods at child yes and child no, respectively, when the question "*Is the gender male?*" is asked.

3.2. Decoding questions

The DTs are built for context-independent phone states. However, the use of phonetic contexts, such as triphones, is well known to improve recognition accuracy. Therefore, we would like to capture phonetic context dependency within the DTs. To handle these, we introduce *"decoding"* questions. They are used to represent contexts such as *context =/b/*or *right context = voiced* for a central phoneme/ah/.

Since different paths during Viterbi decoding refer to different triphone contexts,[b] it is desired that the leaf-values represent $P(\boldsymbol{X}|q, a = Type)$ where *type* is the phonetic context. Therefore, the question is selected and subsequent split is achieved differently as shown in Figure 3. First, only the true samples are required to answer the question and the false samples are propagated to both child nodes. Second, the true samples for one child node are also propagated to the other child node as false samples. Therefore, the total number of samples at both child nodes remains the same. Note that child nodes created as a result of decoding questions have leaf-values of the form:

$$\frac{P(q, a = Type|\boldsymbol{X})}{P(q, a = Type)} \propto P\left(\boldsymbol{X}|q, a = Type\right). \tag{9}$$

The likelihood increase ΔL now is computed as Equation (10) and is directly comparable to Equation (7).

$$\Delta L = N_T^y log\frac{N_T^y}{N_{all} \cdot p_y} + N_T^n log\frac{N_T^n}{N_{all} \cdot p_n} - N_T log\frac{N_T}{N_{all} \cdot p} \tag{10}$$

where p_y and p_n are *prior* probabilities at yes and no nodes, respectively, satisfying $p_y + p_n = p$. These probabilities are different and represent joint prior probability of the true class and the context.

The decoding questions untying a state of the phoneme according to the context. This untying takes place after significant splitting based on normal acoustic questions and therefore there is more effective data sharing across different context classes. For example, a DT model trained for the third state of the phoneme/ah/ resulted in 10,000 leaves while there were only 100 different contexts for the same state of the phoneme/ah/in the GMM baseline system. The DT models have 10 times effective data sharing in this case.

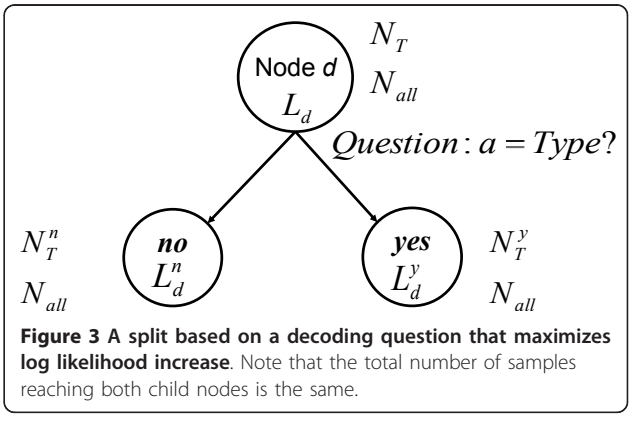

Figure 3 A split based on a decoding question that maximizes log likelihood increase. Note that the total number of samples reaching both child nodes is the same.

During training, the phonetic contexts are determined for the decoding questions from the forced alignments of the training data. At recognition time, the contexts are obtained from the decoding network.

A problem with computing acoustic likelihoods using DTAMs is that the hard yes/no decisions made at various nodes in the tree may lead to big changes in likelihoods. This results in a step likelihood function that is unsuitable for the large variability encountered in speech. A forest comprising of more than one DT, which can alleviate this problem, is explained in the next section.

4. Forest models

A forest[c] is defined as a mixture of DTs. Mixture models benefit from the smoothing property of ensemble methods. The likelihood of a sample $\boldsymbol{X}$ given a forest is computed as:

$$P(\boldsymbol{X}|q) = \sum_j W_j \cdot P(\boldsymbol{X}|T_j^l) \tag{11}$$

where $P(\boldsymbol{X}|T_j^l)$ is provided by one of the leaf-values of the jth tree in the forest and W_j is the corresponding weight. A number of different ways in which a forest can be realized are presented in the following sections.

4.1. Acoustic partitioning

We can achieve partitioning of the acoustic space using a single DT and then create a DTAM for each partition. This technique has an advantage in that the model size does not increase with the number of DTs as is the case with ensemble methods such as bagging [9,10]. The training is formulated in such a way that the weights W_j represent the prior probability $P(T_j|\textit{true class})$. In subsequent expectation maximization EM [10] iterations, the weights W_j and the leaf-values are re-estimated. The algorithm is as follows:

(1)*Initialize the DT components* DT_k *by randomly assigning data points to each component, k, and setting* $W_k = 1/N$ *where N is the number of DT components.*

(2)*Train individual DT components by considering only the assigned samples as true samples and all other samples as false.*

(3)*For every data point* $\boldsymbol{X}_i$*, compute* $L(\boldsymbol{X}_i|DT_k)$ *using individual DT components* DT_k*. Choose* DT_k *that maximizes* $L(\boldsymbol{X}_i|DT_k)$ *and assign the sample to that component.*

(4)*Update* W_k *as:* W_k=(*The number of true samples assigned*))(*Total number of true samples*)

(5)*Compute leaf values for each component using the assigned dataset.*

(6)*Go to* (*3*).

4.2. Speaker clustering

A statistical speaker clustering approach (such as [11]) is used to create a number of clusters and a different tree is trained for each cluster. Specifically, four clusters (two for each gender) are used in this study. Training data from only one specific cluster is used to train the tree for this cluster. This formulation results in the weights W_j representing the posterior probability of the jth cluster. These probabilities are computed separately at the time of decoding for each frame computed using the speaker cluster derived models.

4.3. Multiple representations

A forest can also consist of trees constructed from different data representations, such as different acoustic feature sets. In this study, we have explored Mel cepstrum modulation spectrum (MCMS) [12] features together with MFCC features in the context of a forest. The motivation for using MCMS features is that they emphasize different cepstral modulation frequencies as opposed to first- and second-order derivative features that only emphasize modulation frequencies around 15 Hz. The weights of these components can be learnt at the time of training using the EM algorithm.

Another approach explored in this study is to use both representations together in a single DT. This concatenated representation may not work for GMMs owing to correlation and increased dimensionality as shown in [3]. An advantage of DTAMs is that they do not impose any restriction on the distribution of feature vectors.

5. Experiments and results

Various configurations of training DTAMs and computing acoustic likelihoods at the time of decoding were evaluated on the 5k ARPA Wall Street Journal (WSJ) task. Specifically, we have used SI-84 training material from WSJ0 corpus. There are over 7000 utterances in this training database from 84 different speakers. For testing, we have used the non-verbalized 5k closed testset used in the November 1992 ARPA WSJ evaluation. There are 330 utterances from 8 different speakers in this test database.

5.1. GMM-based baseline systems

A baseline system was setup following [7]. An HMM-based speech recognizer with GMMs was created as a baseline system using HTK V3.4 [13]. The states of the HMM corresponded to cross-word triphones. All triphones had a strict left-to-right topology with three states. A separate DT was constructed for each state of each central phone to tie triphone states in a number of equivalence classes. As a result of clustering, there were

around 12000 physical HMM states and 2753 distinct state PDFs. Each state PDF was associated with 8-component (16 for silence) GMM densities and each component was characterized by a mean vector and a diagonal covariance matrix. This resulted in 1.74M parameters in the GMM system. MFCCs and their first and second derivatives were used for the 39-dimensional vector representation of speech signal every 10 ms. A bigram language model was used for decoding.

The above setting is a standard one for the WSJ evaluation. We also created a GMM-based system with four components per mixture (eight for silence) to make the number of parameters similar to that of the proposed DTAM systems.

5.2. DTAM system

Most of the system components including the dictionary, language model, HMM topology, and MFCC representation were kept exactly the same as the baseline. The decoding was also run exactly the same as the baseline except that the observation likelihoods $P(\boldsymbol{X}|state)$ were computed from the DTAMs instead of GMMs. In each DTAM system, there are only as many DTs as there are monophone states, even in the triphone DTAM case. In the latter systems context-dependent acoustic likelihoods were provided based on the answers to the phonetic context decoding questions. This context information is derived at decoding time.

The number of parameters in DTAM systems is determined by the total number of nodes in DTAMs. These parameters are (a) question thresholds and (b) leaf-values at leaf nodes. As mentioned in Section 2, bottom-up pruning is applied to the trees in order to avoid over-fitting and improve the robustness against unseen data. However, no pruning was applied in the experiments since the model size without any pruning was already much smaller compared to the GMM system.

5.3. Effects of high level information in acoustic models

As shown in Section 3, high-level information such as gender or contexts can be directly incorporated into DTAMs using equal or decoding questions.

Table 1 shows the performance in terms of word error rates for monophone and triphone DTAMs. We can see that context information significantly improves the performance of DTAM systems as expected. 43.7% relative error rate reduction was achieved with triphone models. It is shown in Table 1 that inclusion of the gender information provides 7.7% relative improvement. This improvement is of the same order as that presented in [7] for the same task using GMMs. However, this was achieved in [7] using 50% more parameters for the gender-dependent system compared to a 0.5% increase in the proposed system.

Table 1 Word error rate (%)and the number of parameters of the proposed DTAM systems and conventional GMM systems on the 1992 WSJ non-verbalized 5K closed-test set

System	% WER	Number of parameters
DTAM monophone	22.9	451k
DTAM triphone	12.9	766k
DTAM triphone with gender information	11.9	770k
GMM triphone 4 components/mixture	10.1	870k
GMM triphone 8 components/mixture	7.5	1740k

We used the sample mean of data arriving at a node as the threshold value in creating DTAMs for all the experimental results presented. The word error rate of an equivalent triphone DTAM system was 12.7% when an exhaustive search was made for the threshold, compared to 12.9%. This shows that using the mean of the data as the threshold achieves performance similar to that of an exhaustive search. The method using the sample mean has the advantages of simplicity and meaningful interpretation if speaker adaptation is to be applied.

The context-dependent GMM system with the standard setting (1740k parameters) achieved higher performance than the proposed DTAM systems. However, the difference in the performance between the GMM and DTAM systems became small when the numbers of parameters were similar. The proposed context-dependent DTAMs are highly compact compared to GMMs. Unlike the state-tying mechanism in the GMM setup, contexts in DTAMs are untied only after significant acoustic splitting has taken place, generally at depths 4 and lower. This results in effective data-sharing across various context classes. The difference in the number of parameters between monophone and triphone DTAM systems shows that nearly one-third of the triphone system questions are context questions. It should also be noted that for DTAMs the computational complexity of likelihood computation is only logarithmic. Therefore, as long as the number of active nodes during decoding is kept comparable to the GMM system, DTAMs prove to be much faster compared to GMMs. A similar observation was made in [4] where the number of vector operations required for DTAMs was only 1/16 of that of GMMs for similar accuracy.

One advantage of DTAMs is that feature usage can be easily analysed, unlike GMMs. Table 2 shows the most dominant features used in triphone context dependent DTAMs without gender information. We can see from this table that the dominant feature changes depending on the node depth in DTs. MFCC static features, their first derivatives, right context and left context features

Table 2 The most dominant feature over depth in the DTAM triphone system without gender information

Depth range	Most dominant feature
1-5	MFCC static features
6-7	MFCC first derivatives
8-13	Right context
≥ 14	Left context

are asked in order of traveling down the tree. Figures 4 and 5 show feature-usage distributions over all features asked in triphone DTAMs without gender information for all and vowel classes, respectively. The usage was counted for MFCC features, their dynamic features, right and left contexts. It can be seen from these figures that there are no big differences in the feature-usage distributions for vowel class compared with that for all classes.

5.4. Forest models

Table 3 shows the % WER of various forest DTAMs. Triphone systems with 2 or 4 trees in the table used 2 or 4 DT components to make a forest for each HMM state. From the table, we can see that a forest based on acoustic partitioning achieves the best performance among the MFCC systems. The number of parameters in this forest model is similar to that of a single DT. Therefore, it has no computation or memory overhead at the time of decoding. However, training of the forest required more computation since an iterative estimation of tree weights and their contributions has to be performed.

A forest model with speaker clustering shows improvement over a single DT whose performance is presented in Table 1 but not over a model with acoustic partitioning. One possible reason for this is that cluster

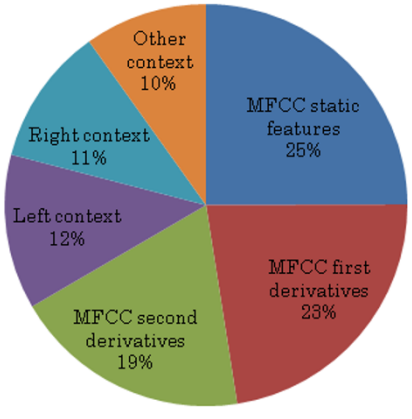

Figure 4 Distribution of feature usage (all classes) in the DTAM triphone system without gender information.

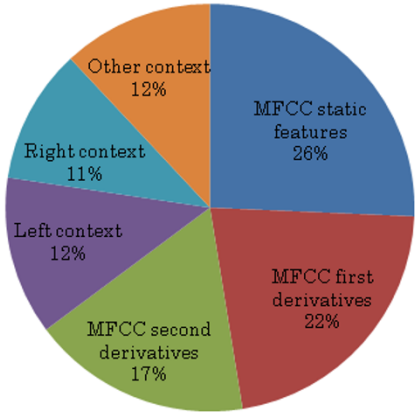

Figure 5 Distribution of feature usage (vowel classes) in the DTAM triphone system without gender information.

weights have to be estimated at the time of decoding. This estimation is prone to mismatch between training and test data. Moreover, the same weights are used for all the trees (phonemes). It is also interesting to see that this performance is similar to that of a gender-dependent system as shown.

A multiple representation forest performs better than both of the individual representation trees (see the first row in Table 1 and the third row in Table 3). It also performs better than the tree obtained using the concatenated representation. The number of parameters is now almost doubled.

Concatenated representations can be used in the DTAM framework although components of the representation are correlated. The resulting system has an even smaller number of parameters and improved performance over individual systems.

6. Conclusions

Various methods for creating DTAMs in speech recognition have been presented in this article. Techniques for training DTs as well as acoustic likelihood computation have been presented for this purpose.

Unordered information such as gender and context was integrated in the acoustic models using equal and decoding questions. The capability of DTAMs to consistently handle both unordered and ordered information makes the data sharing more efficient than in the GMM framework. Consider a hypothetical example of a phoneme where the acoustic signal does not change so much with gender. In the case of GMM, the data are divided into male and female classes. Then, acoustic models for the phoneme are separately trained for each class regardless of no significant acoustic difference between two genders. In DTAMs, a question about gender will be asked after significant splitting based on

Table 3 WER (%) and the number of parameters of triphone forest DTAM systems on the 1992 WSJ non-verbalized 5K closed-test set

System	Number of trees	% WER	Number of parameters
Non-forest (MFCC)	1	12.9	766k
Non-forest with gender information (MFCC)	1	11.9	770k
Non-forest (MCMS)	1	13.3	798k
Non-forest (MCMS + MFCC concatenated)	1	12.5	707k
MCMS + MFCC	2	10.7	1500k
Acoustic partitioning (MFCC)	4	10.9	747k
Speaker clustering (MFCC)	4	11.9	806k

normal acoustic questions. Therefore, DTAMs have more effective data sharing across gender.

Several ways of realizing a forest of DTs were presented and evaluated. A forest based on acoustic partitioning achieved the best performance among the MFCC systems explored in this study. Although this performance was not as good as that of GMMs, several advantages of using DTAMs have been highlighted. These advantages include (a) compactness, (b) computational simplicity, (c) ability to effectively incorporate unordered information, and (d) effectiveness with multiple representations regardless of dimensionality and distribution. We are investigating more techniques to make DT acoustic models as robust and accurate as GMMs while maintaining these advantages. They include techniques (a) employing vector-valued questions at various nodes in the tree, (b) growing one big single tree for all classes leading to even better data sharing and discrimination among classes, and (c) making soft decisions at various nodes. The findings of these experiments will be reported in the future.

Endnotes

[a]Part of this study was presented at Interspeech 2009 [6]. [b]We use cross-word, context-dependent expansion of word networks. [c]There have been some recent applications of decision tree forests to speech recognition, for example, Chen and Zhao explored a forest approach based on overlapped speaker clustering to improve a GMM-based phone recognizer and a recurrent neural network (RNN)-based frame classifier [9].

Abbreviations

ASR: automatic speech recognition; DT: decision tree; DTAM: decision tree-based acoustic model; EM: expectation maximization; GMM: Gaussian mixture model; HMM: hidden Markov model; MCMS: Mel cepstrum modulation spectrum; MFCC: Mel frequency cepstral coefficient; WSJ: Wall Street Journal.

Acknowledgements

The authors thank Dr. Remco Teunen, Google Inc., USA, for useful discussions with him and his contributions to software tools of DTAM training, and Mr. Yusuke Shinohara, Knowledge Media Laboratory, Corporate Research and Development Center, Toshiba Corp., Japan, for his assistance in preparing experiments.

Author details

[1]Toshiba Corporate R&D Center, 1, Komukai Toshiba, Saiwai, Kawasaki 212-8582, Japan [2]IBM Research Lab., 4 Block C, Institutional Area, Vasant Kunj, New Delhi 110070, India

Competing interests

The authors declare that they have no competing interests.

Received: 21 April 2011 Accepted: 17 February 2012
Published: 17 February 2012

References

1. R Cole (ed.), *Survey of the State of the Art in Human Language Technology*. (Cambridge University Press, New York, 1997)
2. JT Foote, Decision-tree probability modeling for HMM speech recognition. *Ph.D. thesis*. (Brown University, Providence, USA, 1993)
3. R Teunen, M Akamine, HMM-based speech recognition using decision trees instead of GMMs. *Proceedings of Interspeech*. (Antwerp, Belgium, 2007), pp. 2097–2100
4. J Droppo, ML Seltzer, A Acero, YB Chiu, Towards a non-parametric acoustic model: an acoustic decision tree for observation probability calculation. *Proceedings of Interspeech*. (Brisbane, Australia, 2008), pp. 289–292
5. J Ajmera, M Akamine, Speech recognition using soft decision trees. *Proceedings of Interspeech*. (Brisbane, Australia, 2008), pp. 940–943
6. J Ajmera, M Akamine, Decision tree acoustic models for ASR. *Proceedings of Interspeech*. (Brighton, UK, 2009), pp. 1403–1406
7. PC Woodland, JJ Odell, V Valtchev, SJ Young, Large vocabulary continuous speech recognition using HTK. in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP1994), vol. 1. (Adelaide, Australia, 1994), pp. 125–128
8. L Breiman, JH Friedman, RA Olshen, CJ Stone, *Classification and Regression Trees*. (Chapman & Hall, New York, 1984)
9. X Chen, Y Zhao, Data sampling ensemble acoustic modelling in speaker independent speech recognition. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP2010)*. (Dallas, USA, 2010), pp. 5130–5133
10. OR Duda, DG Stork, *Pattern Classification*, 2nd edn. (John Wiley & Sons, 2001), Hoboken, NJ, USA)
11. J Ajmera, C Wooters, A robust speaker clustering algorithm. *Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU2003)*. (Virgin Islands, USA, 2003), pp. 411–416
12. V Tyagi, I McCowan, H Misra, H Bourlard, Mel-cepstrum modulation spectrum (MCMS) features for robust ASR. *Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU2003)*. (Virgin Islands, USA, 2003), pp. 399–404
13. S Young, G Evermann, M Gales, T Hain, Dan Kershaw, X Liu, G Moore, J Odell, D Ollason, D Povey, V Valtchev, P Woodland, *The HTK Book, Revised for Version 3.4*. (The University of Cambridge, UK, 2006)

doi:10.1186/1687-4722-2012-10
Cite this article as: Akamine and Ajmera: **Decision tree-based acoustic models for speech recognition.** *EURASIP Journal on Audio, Speech, and Music Processing* 2012 **2012**:10.