INTERSPEECH
2009 BRIGHTON

Decision Tree Acoustic Models for ASR

Jitendra Ajmera, Masami Akamine

Toshiba Corporate Research and Development Center,
Kawasaki, Japan

jitendra.ajmera@toshiba.co.jp, masa.akamine@toshiba.co.jp

Abstract

This paper presents a summary of our research progress using decision-tree acoustic models (DTAM) for large vocabulary speech recognition. Various configurations of training DTAMs are proposed and evaluated on wall-street journal (WSJ) task. A number of different acoustic and categorical features have been used for this purpose. Various ways of realizing a *forest* instead of a single tree have been presented and shown to improve recognition accuracy. Although the performance is not shown to be better than Gaussian mixture models (GMMs), several advantages of DTAMs have been highlighted and exploited. These include compactness, computational simplicity and ability to handle unordered information.

Index Terms: speech recognition, decision trees, hidden Markov model (HMM)

10.21437/Interspeech.2009-432

1. Introduction

Gaussian mixture models (GMMs) are used to model state probability density functions (PDFs) in most hidden Markov model (HMM) based automatic speech recognition (ASR) systems. These state PDFs estimate the likelihood of a speech sample X given a particular *state* of the HMM, denoted as $P(X|state)$. The sample X is typically a vector representing speech signal over a short time window, e.g. mel frequency cepstral coefficients (MFCC). Although some other alternatives such as artificial neural networks (ANNs), support vector machines (SVMs) have also been studied for computing the acoustic likelihood $P(X|state)$, GMMs have remained the most researched and successful choice [1].

On the other hand, while decision trees (DT) are powerful statistical tools and have been widely used for many pattern recognition applications, their effective usage in ASR is limited to state-tying prior to building context-dependent GMM densities [2]. Recently some attempts have been made to use DTs for computing the acoustic likelihood instead of GMM [3, 4, 5]. However, only simple tasks such as digit or phoneme recognition have been explored in these works.

DTs are attractive for a number of reasons including their simplicity, interpretability and ability to better incorporate unordered information. If used as acoustic models (DTAMs), they can offer additional advantages over GMMs as: a) they are discriminative models, b) make no assumption about distribution of underlying data, and c) are computationally very simple.

The goal of this research is to therefore explore and exploit DTs for the purpose of large vocabulary speech recognition. Various configuration of training DTAMs have been presented and evaluated in this paper. Section 2 presents an overview of the proposed acoustic models including training. Section 3 presents various ways of realizing *forest* which is shown to be more robust and accurate than single DT. Section 4 presents experimental framework and evaluation of various proposed configurations. Although the observed performance is not better than GMM acoustic models, several advantages of DTAMs are highlighted and discussed in Section 4.

2. Decision Tree Acoustic Models (DTAMs)

As mentioned above, DTAMs are used in this work to model state PDFs estimating $P(X|state)$. A separate binary DT is associated with each *state* of the HMM[1]. Each DT is trained to maximize the discrimination between the *true* samples (data associated with the corresponding HMM *state*) and all other (*false*) samples. In the following discussion, we will use the notation $P(X|true\ class)$ instead of $P(X|state)$.

The parameter estimation process of DTs consists of a growing stage and a bottom-up pruning stage. A binary DT is grown by splitting a node into two child nodes. The training algorithm considers all possible splits and selects the split that maximizes likelihood increase (ΔL) given by Eq. 1. This choice of split is represented in the form of a question such as $\langle x \leq \tau \rangle$ where x is one of the attributes of the data ($x \in X$) and τ is the corresponding threshold. If the number of *true* samples reaching a node is N and the total number of samples (*false* and *true*) is D, ΔL is:

$$\Delta L = N_{yes} \log \frac{N_{yes}}{D_{yes}} + N_{no} \log \frac{N_{no}}{D_{no}} - N \log \frac{N}{D} \qquad (1)$$

where, $N_{yes}(D_{yes})$ and $N_{no}(D_{no})$ are the *true (all)* samples answering the split question $\langle x \leq \tau \rangle$ in *yes* and *no*, respectively as shown in Figure 1.

Since we are dealing with one scalar component of the representation at one time (unlike vector questions in [4]), it is possible to perform an exhaustive search over all possible values of x and τ to find the best question that maximizes ΔL in Eq. 1.

In this work, however, we propose to use *sample mean* of data arriving at a node as the threshold value for each component x. It is shown in Section 4 that it provides similar performance to that of exhaustive search. Also, it is simple to compute and has meaningful interpretation for the task of speaker adaptation.

The process of splitting can be continued as long as there are split-able nodes. For a node to be split-able: a) the node should have a minimum number of *true* samples, and b) the resulting split must satisfy chi-square significance test. When a node cannot be split any further, it is referred to as a leaf-node

[1] Other configuration such as one global DT for all the phonemes [4] has not been explored in this work.

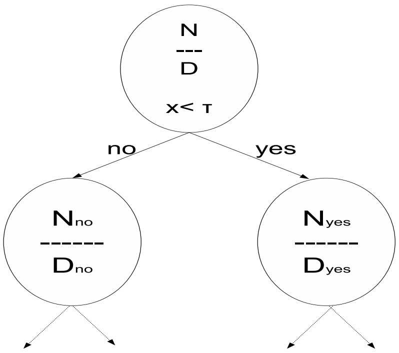

Figure 1: A node in a decision tree is split based on a question such as $\langle x \leq \tau \rangle$ that maximizes likelihood of splitting (Eq. 1).

and its *leaf-value* provides the likelihood of sample X as:

$$P(X|true\ class) \propto \frac{P(true\ class|X)}{P(true\ class)} = \frac{N}{D \cdot p} \quad (2)$$

where $p = P(true\ class)$ is the *prior* probability of the *true* class and is given by the fraction of the *true* samples at the root-node of the tree. This leaf-value is then passed to the Viterbi decoder as acoustic likelihood.

Once a tree is fully grown, a bottom-up pruning is performed in a worst-first fashion to get desired number of nodes in the DT.

2.1. Equal Questions

One of the biggest advantage of DTAMs over GMMs is that they can efficiently embed unordered information such as gender, context etc. in the core model itself. A question of the form $\langle l == Type \rangle$ is used for this purpose where l is one of the attributes (e.g. gender) of the data. There are two ways in which these questions can be implemented. If these equal questions are trained and asked in the same manner as above (Eq. 1, Figure 1), the corresponding leaf-values would represent following:

$$\frac{P(true\ class|X, l = Type)}{P(true\ class)} \propto \frac{P(X, l = Type|true\ class)}{P(l = Type|X)} \quad (3)$$

This is applicable for information such as *gender* where the posterior probability $P(gender = male/female|X)$ can be computed after test data X is observed. The overall likelihood $P(X|true\ class)$ then can be computed as a weighted sum of the gender-based likelihood given by Eq. 3.

2.2. Context Modeling

Since different paths during Viterbi decoding refer to different triphone contexts [2], it is desired that the leaf-values represent $P(X|true\ class, l = Type)$. Therefore, the question is selected and subsequent split is achieved differently as shown in Figure 2. 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. We refer to these types of questions as *decoding questions*. Note that child nodes created as result of decoding questions have *leaf-values* of the form:

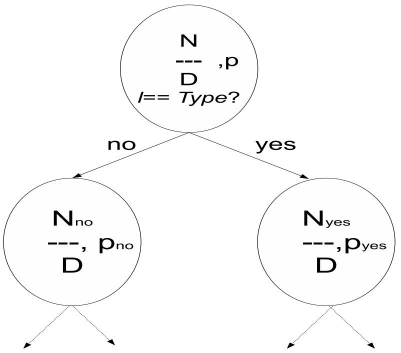

Figure 2: A decoding question. Note that the total number of samples reaching both child nodes is same (D).

$$\frac{P(true\ class, l = Type|X)}{P(true\ class, l = Type)} \propto P(X|true\ class, l = Type) \quad (4)$$

The likelihood increase ΔL now is computed as Eq. 5 and is directly comparable to Eq. 1.

$$\Delta L = N_{yes} \log \frac{N_{yes}}{D \cdot p_{yes}} + N_{no} \log \frac{N_{no}}{D \cdot p_{no}} - N \log \frac{N}{D \cdot p} \quad (5)$$

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

In this work, decoding questions are used to represent context such as $\langle left\ context == /ah/ \rangle$ or $\langle right\ context == voiced \rangle$ resulting in context *untying*. 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, if we define *specific context* by a group of less than 5 phonemes, the 10000 leaves of the tree for the 3rd state of the phoneme */ah/* collectively represent around 1000 specific unique triphone contexts. The corresponding number in our baseline GMM system is less than 100.

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 step likelihood function which is not suitable for large variability encountered in speech. A *forest* comprising of more than one DTs is explained in next section which can alleviate this problem.

3. Forest

A forest is defined as a mixture of decision trees. Mixture models benefit from the smoothing property of ensemble methods. The likelihood of a sample X given a forest is computed as:

$$P(X|true\ class) = \sum_j W_j \cdot P(X|Tree_j) \quad (6)$$

[2] We use Cross-word, context-dependent expansion of word networks.

where, $P(X|Tree_j)$ is provided by one of the *leaf-values* of the j^{th} 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 following subsections.

3.1. Acoustic Partitioning

In this case a partitioning of acoustic space is achieved using one single DT and then a number of DTs are created for each partition as explained in [3]. This technique has the advantage that the model size does not increase with number of DTs as is the case with ensemble methods such as bagging [6]. The training is formulated in such a way that the weights W_j represent the *prior* probability $P(Tree_j|true\ class)$. In subsequent *expectation maximization* EM [6] iterations, the mixture weights W_j and the leaf-values are re-estimated.

3.2. Speaker Clustering

A statistical speaker clustering (such as [7]) is used to create a number of clusters and a different tree is trained for each cluster. Specifically, 4 clusters (2 for each gender) are used in this work. Training data from only one specific cluster is used to train a tree.

This formulation results in the weights W_j representing posterior probability of j^{th} cluster ($P(cluster_j|X)$). These probabilities are computed separately at the time of decoding using clusters models obtained as a result of speaker clustering.

3.3. Multiple Representations

A forest can also consist of trees constructed from different data representations. In this work, we have explored Mel cepstrum modulation spectrum (MCMS) [8] 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 which only emphasize modulation frequencies around 15Hz. The weights of these components can be learnt at the time of training using EM algorithm.

Another approach explored in this work is to use both representations together in single DT. Although, this approach brings improvement in recognition accuracy without increasing model size, the improvement is not as good as with the forest method as shown in Section 4. Another point to note is that this concatenated representation may not work for GMMs due to correlation and increased dimensionality as shown in [3]. This is another advantage of DTAMs that they do not impose any restriction on the distribution of feature vectors.

4. Experiments and Results

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

4.1. Baseline Setup

A baseline system was setup as explained in [2]. An HMM based speech recognizer with GMM state PDFs was created using HTK [9]. The states of the HMM correspond to cross-word triphones. All triphones have 3 state emitting states and a strict left-to-right topology. A separate decision tree is constructed for each state of each base class with the goal of grouping triphone states into a number of equivalence classes. As a result of clustering, there are around 12000 physical HMMs and 2753 distinct state PDFs.

MFCCs and their first and second derivatives for the 39-dimensional vector representation of speech signal every 10ms. A bi-gram language model was used for decoding.

4.2. DTAM Setup

Most of the system components like dictionary, language model, HMM topology, MFCC representation etc. have been kept exactly the same as the baseline.

The decoding also runs exactly the same as baseline except that the observation likelihoods $P(X|state)$ are computed using DTAMs instead of GMMs. Note that although proposed system is context-dependent, there are only as many DTs as there are *monophone* states. Each DT, in turn, can provide context-dependent acoustic likelihood depending on the answers to context questions. The context information is derived at the time of decoding.

4.3. Memory and Computational Requirements

In GMM system, 2753 state PDFs are associated with 8-component (16 for silence) GMM densities and each component is characterized by a mean vector and a diagonal covariance matrix. This means that there are 1.74M parameters in the GMM system.

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. No pruning was applied since the model size without any pruning was already much smaller compared to GMM system.

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. Similar observation was made in [4] where number of vector operations required for DTAMs was only 1/16 of that of GMMs for similar accuracy.

Performance in terms of percentage recognition accuracy for various DT configurations and corresponding model-size are presented in Table 1. Following can be observed from these results:

4.4. Discussions

1. As expected, context information helps DTAMs (Row 1 and 2) systems the most. The difference in number of parameters between Monophone and Triphone system shows that nearly one-third of the questions are context questions.

2. Context-dependent DTAMs are highly compact compared to GMMs (Rows 2 and 10). Unlike the *state-tying* mechanism in GMM setup, contexts in DTAMs are *untied* only after significant acoustic splitting has taken place (generally depths 4 and lower). This results in effective data-sharing across various context classes.

3. Using *mean* of the data as threshold achieves similar performance to that of an exhaustive search (Rows 2 and 3)

Table 1: *Percentage Accuracy of different Methods.*

	System	% Accuracy	Number of Parameters
1	DTAM monophone	77.1	451k
2	DTAM **triphone** (Section 2.2)	87.1	766k
3	DTAM triphone (**Exhaustive** search)	87.3	855k
4	DTAM triphone MFCC with **Gender** (Section 2.1)	88.1	770k
5	DTAM triphone Forest **Acoustic Partioning** (Section 3.1)	**89.1**	747k
6	DTAM triphone Forest **Speaker Clustering** (Section 3.2)	88.1	806K
7	DTAM triphone Forest **MCMS** and MFCC (Section 3.3)	89.3	1.5M
8	DTAM triphone MCMS	86.7	798K
9	DTAM triphone MCMS and MFCC (**concatenated**)	87.5	707K
10	GMM triphone baseline	**92.5**	1.74M

but has the advantage of simplicity. Exhaustive search can potentially pick fine details of the training data and lead to over-training. Moreover, *mean* has meaningful interpretation, especially, for the speaker adaptation step.

4. Inclusion of the gender information provides 7.7% relative improvement (Row 4) which is of the same order as presented in [2] for exactly the same task using GMMs. However, this was achieved in [2] using 50% more parameters for the gender-dependent system compared to 0.5% increase in the proposed system.

5. A forest based on acoustic partitioning achives best performance among various configurations explored in this work. The number of parameters in this forest is similar to that of a single decision tree. Therefore it has no computation or memory overhead at the time of decoding. However, training of a forest required more computation since an iterative estimation of tree weights and their contributions have to be performed.

6. Clustering (Row 6) shows improvement over a single DT (Row 2) but not over a random forest (Row 5). One possible reason for this is that cluster weights $P(cluster_j|X)$ have to be estimated at the time of decoding. This estimation is prone to mismatch between training and test data. Moreover, same weights are used for all the trees (phonemes). It is also interesting to see that this performance is similar to that of gender-dependent system (Row 4).

7. A multiple representation forest (Section 3.3, Row 7) performs better than both of the individual representation trees (Rows 2 and 8). It also performs better than the tree obtained using concatenated representation (Row 9). The number of parameters is now almost doubled.

8. Concatenated representation (Row 9) can be used in DTAM framework although components of the representation are correlated. The resulting system has even smaller number of parameters and improved performance over individual systems (Rows 2 and 8).

9. None of the DTAM configuration can achieve as good performance as that of GMMs (Row 10). We are looking at several ways in which the performance of DTAMs can be improved such as 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 future.

5. Conclusions

Various methods of using decision tree based acoustic models in speech recognition have been presented in this paper. Techniques for training decision trees as well as acoustic likelihood computation have been presented for this purpose. Unordered information such as context and gender were integrated in the acoustic models and analysis showed that such information is better handled in DTAMs than in GMM framework. Several ways of realizing *forest* of decision trees were explained and evaluated. Forest based on acoustic partitioning achieves the best performance among various configurations explored in this work. Although this performance was not as good as 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 decision tree acoustic models as robust and accurate as GMMs while maintaining these advantages.

6. References

[1] Cole, R. et al. (eds), "Survey of the State of the Art in Human Language Technology", Cambridge University Press, New York, 1997.

[2] Woodland, P.C., Odell, J.J., Valtchev, V. and Young, S.J., "Large Vocabulary Continuous Speech Recognition using HTK", ICASSP 1994, 125–128, 1994.

[3] Teunen, R and Akamine, M., "HMM based speech recognition using decision trees instead of GMMs", Interspeech 2007, 2097–3000, 2007.

[4] Droppo, J., Seltzer, M., Acero, A. and Chiu, Y.-H., "Towards a non-parametric acoustic model: An acoustic decision tree for observation probability calculation" , Interspeech 2008, 289–292, 2008.

[5] Ajmera, J. and Akamine, M., "Speech recognition using Soft decision trees", Interspeech 2008, 940–943, 2008.

[6] Duda, O.R., Hart, P. E., and Stork, D. G., "Pattern Classification", John Wiley & Sons, Second Edition, 394–453, 2001.

[7] Ajmera, J. and Wooters, C., "A robust speaker clustering algorithm", ASRU, 411-416, 2003 .

[8] Tyagi, V., McCowan, I., Misra, H. and Bourlard, H., "Mel-Cepstrum Modulation Spectrum (MCMS) features for robust ASR", ASRU 2003, 399–404, 2003.

[9] Young, S., Kershaw, D., Odell, J., Ollason, D., Valtchev, V., and Woodland, P., The HTK Book (for HTK Version 3.0), Cambridge University, 2000.